

AD-A169 882

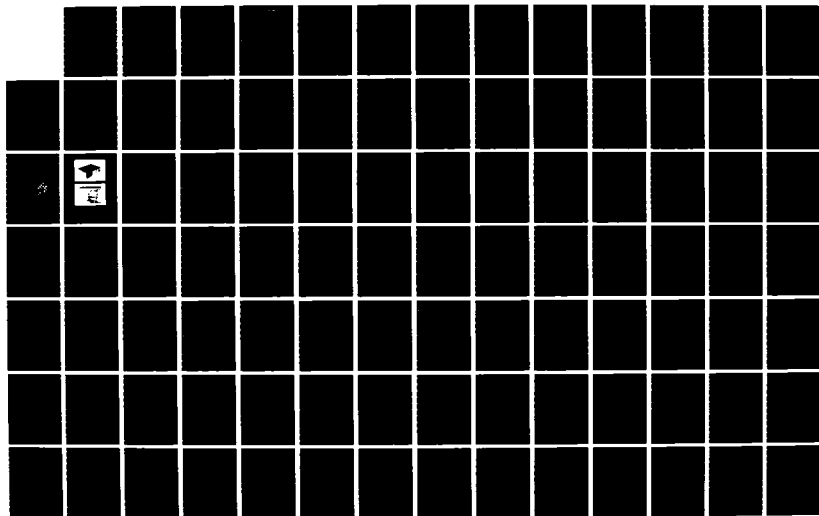
OPTICAL PROCESSING(U) DAYTON UNIV OH RESEARCH INST  
S C GUSTAFSON 31 DEC 85 UDR-TR-85-148 N00014-85-K-0479

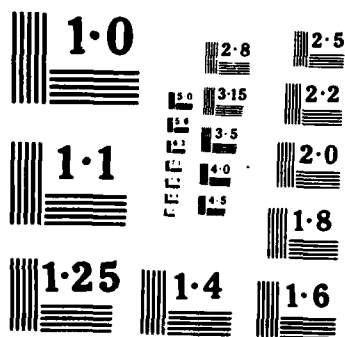
1/6

UNCLASSIFIED

F/G 9/2

NL





OPTICAL PROCESSING

AD-A169 882

Editor  
Steven C. Gustafson, Ph.D.  
University of Dayton  
Research Institute  
Dayton, OH 45469

December 31, 1985

Interim Report

DTIC FILE COPY

Approved for public release; distribution unlimited.

Prepared for:

Office of Naval Research (ONR)  
800 North Quincy Street  
Arlington, VA 22217-5000

JUL 21 1986

E

86 7 10 232

## **DISCLAIMER NOTICE**

**THIS DOCUMENT IS BEST QUALITY  
PRACTICABLE. THE COPY FURNISHED  
TO DTIC CONTAINED A SIGNIFICANT  
NUMBER OF PAGES WHICH DO NOT  
REPRODUCE LEGIBLY.**



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

## REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION University of Dayton Research Institute	6b. OFFICE SYMBOL (If applicable) KL464	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research (ONR)	
6c. ADDRESS (City, State and ZIP Code) 300 College Park Dayton, Ohio 45469		7b. ADDRESS (City, State and ZIP Code) 800 N. Quincy St. Arlington, VA 22217-5000	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER Contract No. N00014-85-K-0479	
8c. ADDRESS (City, State and ZIP Code)		10. SOURCE OF FUNDING NOS.	
		PROGRAM ELEMENT NO.	TASK NO.
11. TITLE (Include Security Classification) Optical Processing		PROJECT NO.	WORK UNIT NO.
12. PERSONAL AUTHOR(S) Steven C. Gustafson, ed.			
13a. TYPE OF REPORT Interim	13b. TIME COVERED FROM 1 Jun 85 TO 31 Dec 85	14. DATE OF REPORT (Yr., Mo., Day) 31 December 1985	15. PAGE COUNT
16. SUPPLEMENTARY NOTATION Supported by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization (SDIO/IST) and administered through the Office of Naval Research (ONR) under Contract No. N00014-85-K-0479			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB. GR.	
		optical computing, optical processing	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>This program covered a broad range of basic research in the optical processing and computing areas. A majority of the effort was carried out by 13 subcontractors (seven universities and six industry organizations). The performing organizations and their technical contributions were as follows: (1) Aerodyne Research Inc., optical parallel 2-D neighborhood processor and optical processor assessment technique; (2) University of Alabama in Huntsville, high accuracy with moderately accurate components and optical fredkin gate architectures; (3) Battelle Columbus Laboratories, integrated optical threshold computing, pipelined polynomial processor, and all-optical analog/digital converter; (4) BDM Corporation, adaptive optical associative memory model with attention; (5) California Institute of Technology, effectiveness of parallelism and connectivity in optical computers; (6) University of California, Irvine, optical systolic array processing using an integrated acoustooptic module; (7) University of Dayton Research Institute, optical threshold elements and networks, holographic threshold processors,</p>			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input checked="" type="checkbox"/> DTIC USERS <input type="checkbox"/>		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Edward J. Wegman, Ph.D.		22b. TELEPHONE NUMBER (Include Area Code) 202-969-4310	22c. OFFICE SYMBOL ONR

adaptive matched spatial filtering, and coherence theory in optical computing; (8) Carnegie-Mellon University, time-varying optical processing for sub-pixel targets, optical Kalman filtering, and adaptive matched filtering; (9) Georgia Institute of Technology, optical degrees of freedom, ultra-short optical pulses, number representations, content-addressable-memory processors, and integrated optical Givens rotation devices; (10) Probe Systems Inc., optical J-K flip-flop analysis and interfacing for optical computers; (11) RGB Associates, matrix multiplication algorithms and limits of incoherent optical computers; (12) Science Applications International Inc., architecture for machine vision with sensor fusion, pattern recognition functions, and neural net implementations; (13) University of Southern California, optical computing algorithms, architectures, and components; (14) Stanford University, dynamic optical interconnections, advantages and architectures.

Account for	
THIS NAME	<input checked="" type="checkbox"/>
THIS JOB	<input type="checkbox"/>
Unchanged	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



## TABLE OF CONTENTS

	<u>Page No.</u>
PROJECT SUMMARY . . . . .	iv
ABSTRACT . . . . .	vi
ADAPTIVE OPTICAL COMPUTING . . . . .	1-1
Ravindra A. Athale, Principal Investigator	
The BDM Corporation	
OPTICAL COMPUTING STRATEGIES . . . . .	2-1
Richard Barakat, Principal Investigator	
RGB Associates, Inc.	
TIME-VARYING OPTICAL DATA PROCESSING FOR SDI . . . . .	3-1
David Casasent and B.V.K. Vijaya,	
Principal Investigators	
Carnegie-Mellon University	
NEW METHODS FOR OPTICAL COMPUTING . . . . .	4-1
H. J. Caulfield, Principal Investigator	
Joseph Shamir	
Mostafa Abushagur	
Stephen Welsted	
The University of Alabama in Huntsville	
A-O COMPUTING STUDY . . . . .	5-1
Jerry L. Erickson, Principal Investigator	
PROBE Systems, Inc.	
DYNAMIC OPTICAL INTERCONNECTIONS . . . . .	6-1
J. W. Goodman and L. Hesselink,	
Principal Investigators	
Stanford University	

## TABLE OF CONTENTS (con't)

<b>THRESHOLDING AND OPTICAL COMPUTING . . . . .</b>	<b>7-1</b>
Steven C. Gustafson, Principal Investigator	
Steven L. Cartwright	
David L. Flannery	
Gordon R. Little	
John S. Loomis	
L. Maugh Vail	
University of Dayton Research Institute	
 <b>INTEGRATED-OPTICAL DEVICES FOR NONLINEAR COMPUTATION . . .</b>	 <b>8-1</b>
Richard P. Kenan and Carl Verber,	
Principal Investigators	
Battelle, Columbus Laboratories	
 <b>THEORETICAL INVESTIGATION OF THE EFFECTIVE UTILIZATION . .</b>	 <b>9-1</b>
<b>OF PARALLELISM AND CONNECTIVITY IN OPTICAL COMPUTERS</b>	
Demetri Psaltis and Yaser S. Abu-Mostafa,	
Principal Investigators	
California Institute of Technology	
 <b>2-D NEIGHBORHOOD PROCESSOR . . . . .</b>	 <b>10-1</b>
Roger S. Putnam and H. J. Caulfield,	
Principal Investigators	
Aerodyne Research, Inc.	
 <b>RESEARCH ON OPTICAL COMPUTING ALGORITHMS AND ARCHITECTURES</b>	 <b>11-1</b>
William T. Rhodes, Project Director	
and Principal Investigator, and	
Thomas K. Gaylord, Principal Investigator	
Georgia Institute of Technology	
 <b>MACHINE VISION . . . . .</b>	 <b>12-1</b>
William Stoner, Principal Investigator	
Science Applications International Corporation	
 <b>OPTICAL SYSTOLIC ARRAY PROCESSING USING A NOVEL . . . . .</b>	 <b>13-1</b>
<b>INTEGRATED ACOUSTO-OPTIC MODULE</b>	
Chen S. Tsai, Principal Investigator	
University of California in Irvine	
 <b>OPTICAL COMPUTING COMPONENTS: FUNDAMENTAL ISSUES . . . .</b>	 <b>14-1</b>
Armand R. Tanguay, Jr. Principal Investigator	
University of Southern California	

## PROJECT SUMMARY

The work reported here was supported by the Innovative Science and Technology Office (IST) of the Strategic Defense Initiative Organization (SDIO) and was administered through the Office of Naval Research (ONR) under Contract No. N00014-85-K-0479. The period of performance was 1 June to 31 December 1985. The project monitors were Dr. James Ionson and Dr. Dwight Dustin at SDIO/IST and Dr. Edward Wegman at ONR. The project science and technology agents were Dr. Keith Bromley and Mr. William Miceli at the Naval Ocean Systems Center (NOSC). The project manager at the University of Dayton (prime contractor) was Dr. Eugene Gerber, and the principal investigator was Dr. Steven Gustafson. The program technical director was Dr. H. John Caulfield at the University of Alabama in Huntsville.

The technical program covered a broad range of basic research in the optical processing and computing areas. Technical presentations were made by the University of Dayton and the thirteen subcontractors at special meeting sessions in San Diego, CA on 21 August and in Washington, DC on 14 October 1985. For the final report, each organization was asked to submit a technical abstract, a summary (including objectives, description of work performed and results, and conclusions and recommendations), and a technical discussion which could consist of papers prepared for publication on the program. The University of Dayton prepared ten candidate papers for publication through the program, and at least this number were prepared by the subcontractors. Thus, as an intense basic research effort, the program was clearly successful.

## ABSTRACT

This program covered a broad range of basic research in the optical processing and computing areas. A majority of the effort was carried out by 13 subcontractors (seven universities and six industrial organizations). The performing organizations and their technical contributions were as follows: (1) Aerodyne Research Inc., optical parallel 2-D neighborhood processor and optical processor assessment technique; (2) University of Alabama in Huntsville, high accuracy with moderately accurate components and optical fredkin gate architectures; (3) Battelle Columbus Laboratories, integrated optical threshold computing, pipelined polynomial processor, and all-optical analog/digital converter; (4) BDM Corporation, adaptive optical associative memory model with attention; (5) California Institute of Technology, effectiveness of parallelism and connectivity in optical computers; (6) University of California, Irvine, optical systolic array processing using an integrated acoustooptic module; (7) University of Dayton Research Institute, optical threshold elements and networks, holographic threshold processors, adaptive matched spatial filtering, and coherence theory in optical computing; (8) Carnegie-Mellon University, time-varying optical processing for sub-pixel targets, optical Kalman filtering, and adaptive matched filtering; (9) Georgia Institute of Technology, optical degrees of freedom, ultra-short optical pulses, number representations, content-addressable-memory processors, and integrated optical Givens rotation devices; (10) Probe Systems Inc., optical J-K flip-flop analysis and interfacing for optical computers; (11) RGB Associates, matrix multiplication algorithms and limits of incoherent optical computers; (12) Science Applications International Inc., architecture for machine vision with sensor fusion, pattern recognition functions, and neural net implementations; (13) University of Southern California, optical computing algorithms, architectures, and components; (14) Stanford University, dynamic optical interconnections, advantages and architectures.

**ADAPTIVE OPTICAL COMPUTING**

**Ravindra A. Athale, Principal Investigator**

**THE BDM CORPORATION**

THE BDM CORPORATION

SUBCONTRACTING ORGANIZATION:

THE BDM CORPORATION  
7915 JONES BRANCH DRIVE  
MCLEAN, VA 22102

PRINCIPAL INVESTIGATOR:

DR. RAVINDRA A. ATHALE

TITLE:

ADAPTIVE OPTICAL COMPUTING

ABSTRACT

A model for optical associative memory is described that incorporates attention. This attention is in the form of increased pre-disposition towards certain memory states. Furthermore, the new formulation of optical associative memory allows for nonlinear processing in the correlation domain, thereby increasing robustness to various errors. Results of computer simulation are presented and the design of an optoelectronic testbed is described.



SUMMARY

A. OBJECTIVES

The concept of associative storage of data is an attractive one for parallel systems, such as optical processors. In an associative data storage, information is retrieved not by location, but by another piece of information that could be distinct from the stored information (hetero-associative) or could be the same as the stored information (auto-associative). The objective of this contract was to look at different formulations of associative memory and find one that is suitable for optical implementation. A further objective of the program was to introduce more versatility into the performance characteristics of the associative data storage.

B. DESCRIPTION OF WORK PERFORMED AND RESULTS

The models of associative memory proposed in the literature involve storing the data in a matrix form by summing the rank-one matrices that are results of outer product operations between the data vectors to be stored and the key vectors that are to be used during retrieval. Such a method produces a delocalized storage of data in which one datum is stored at several locations and several sites collaborate in storing one piece of information. This features results in a system that is very resistant to failure of hardware components. Such a system is also relatively immune to noisy and/or incomplete retrieval vector. Such systems have been under investigation for last 20 years. Recently Psaltis and Farhat have proposed an optical implementation of one particular model of auto-associative memory developed by Hopfield.

A closer inspection of the equations for the Hopfield model revealed that it can be decomposed into a two-step process: (i) compare the input vector with the stored set of data vectors, (ii) calculate a linear superposition of the stored vectors with the similarity measures calculated in

step (i) as the coefficients. This procedure is then iteratively applied with a hardclipping nonlinearity in between to binarize the retrieved vector (the stored vectors are known to be binary). We then proposed a model of associative memory where these two steps are explicitly carried out via two vector-matrix multiplications. This allows one to weight the data vectors differently in the correlation (or inner product) domain as well as introduce a nonlinearity to suppress cross-talk. The nonuniform weighting corresponds to introducing ATTENTION in the associative memory, and hence the name ATTENTIVE ASSOCIATIVE MEMORY.

In this six month period, we performed computer simulations of the basic model of attentive associative memory (AAM). The simulations showed that the AAM is capable of storing vectors that are highly correlated and yet retrieve them without error. It also demonstrated that by nonuniform weighting one can retrieve a vector that is weaker as compared to another vector. An optoelectronic testbed was also designed that requires off-the-shelf components. These results were presented at a post-deadline presentation at the 1985 Annual Meeting of the Optical Society of America. The fabrication of the optoelectronic testbed was completed using internal BDM funds. The testbed performed in a manner consistent with the computer simulations.

#### C. CONCLUSIONS AND RECOMMENDATIONS

It was concluded that an alternate formulation of associative memory that does not involve the delocalized storage via outer products introduces additional flexibility in the operation of the memory. This flexibility can be utilized to incorporate ATTENTION or predisposition in the stored states as well as to introduce additional nonlinearities for cross-talk suppression.

The attentive associative memory can be extended to hetero-associative data storage. This new direction can have important consequences for data base systems. The vector-matrix multiplier required by this architecture can be implemented in a compact optical system by properly

## THE BDM CORPORATION

designed spatial light modulators. An important aspect of the associative memory is the training of the system. A very fruitful direction of research will be the study of different learning mechanisms as applied to attentive associative memory.

THE BDM CORPORATION

ATTENTIVE ASSOCIATIVE MEMORY AND ITS OPTICAL IMPLEMENTATION

RAVINDRA A. ATHALE, HAROLD H. SZU,\* AND CARL B. FRIEDLANDER

THE BDM CORPORATION  
7915 JONES BRANCH DRIVE  
MCLEAN, VA. 22102

\*US NAVAL RESEARCH LABORATORY  
CODE 5709  
WASHINGTON, DC. 20375

ABSTRACT

A mathematical model for incorporating ATTENTION in the conventional associative memory is described. Such a mechanism provides the flexibility of changing rapidly the strengths of the stored states in an associative memory. This Attentive Associative Memory can be implemented optically. Results obtained with computer simulation and an optoelectronic testbed with off-the-shelf components will be discussed.

A. INTRODUCTION

The optical processors have the unique features of massive parallelism, flexible and global interconnection, and ease of performing analog multiplication and addition. Over the years a number of architectures based on Fourier transform and convolution/correlation operation have been developed. In the last two years, neural network models have excited the imagination of the optical processing community as a source of inspiration. The pioneering work of Psaltis and Farhat,<sup>1</sup> and Fisher, Giles, and Lee<sup>2</sup> has been followed by an increased activity as evidenced by the special symposium on Associative Memories and Optics at the 1985 Annual Meeting of the Optical Society of America.<sup>3</sup>

In this letter we will modify the conventional formulation of associative memory to incorporate attention, or increased sensitivity to the presence of a given stored data vector. We will first describe one version of the associative memory formulated in terms of linear algebra, then we will discuss the proposed modification along with its consequences. The results of a computer simulation will be discussed next followed by the design of an optoelectronic testbed and experimental results obtained with it. Finally we will discuss some directions for further work.

B. MATHEMATICAL FORMULATION OF ATTENTIVE ASSOCIATIVE MEMORY

The simplest model of an associative memory, which is designed to store  $N$ -dimensional column vectors, contains two steps: the first step is the recording of the set of  $P$  input vectors in an  $N \times N$  memory matrix via the outer product operation between the input vectors, and the second step is retrieving the data vector from an incomplete and/or noisy version of the vector itself via a vector-matrix multiplication. These two steps are described mathematically in the following equation:<sup>4</sup>

$$\begin{aligned}
 M &= \sum \underline{v}^{(i)} \underline{v}^{(i)T} && \text{RECORDING} \\
 \underline{v} &= M \underline{v}' && \text{RETRIEVAL} \\
 v_j &= \sum M_{jk} v_k' && (1)
 \end{aligned}$$

where  $\underline{v}$  is an N-dimensional column vector, M is an N X N matrix,  $\underline{v}'$  is the imperfect recall vector, and  $\underline{v}^{(i)}$  is one of the stored vectors. The last part of Equation (1) can be rewritten by a substitution for values of  $M_{jk}$  derived from the first part of Equation (1):

$$v_j = \sum (\sum v_j^{(i)} v_k^{(i)}) v_k' \quad (2)$$

The attentive associative memory formulation can be obtained by changing the order of summation Equation (2) and inserting a nonuniformly nonlinear operation after the first summation. The resultant equation is given below:

$$v_j = \sum v_j^{(i)} \lambda(i) [\sum v_k^{(i)} v_k'] \quad (3)$$

This equation states that the imperfect input vector  $\underline{v}'$  is first compared to all the stored vectors in parallel via an inner product, the resultant scalar is transformed using channel-dependant nonlinearity  $\lambda(i)$  and then used as a coefficient in a linear superposition of the corresponding stored vectors. The output is thus an estimate of the stored vector that is closest to the input vector  $\underline{v}'$ . The nonlinear operation  $\lambda(i)$  allows one to suppress spurious correlations and to emphasize the similarity of the input with a selected vector (i.e. focusing attention on that particular vector).

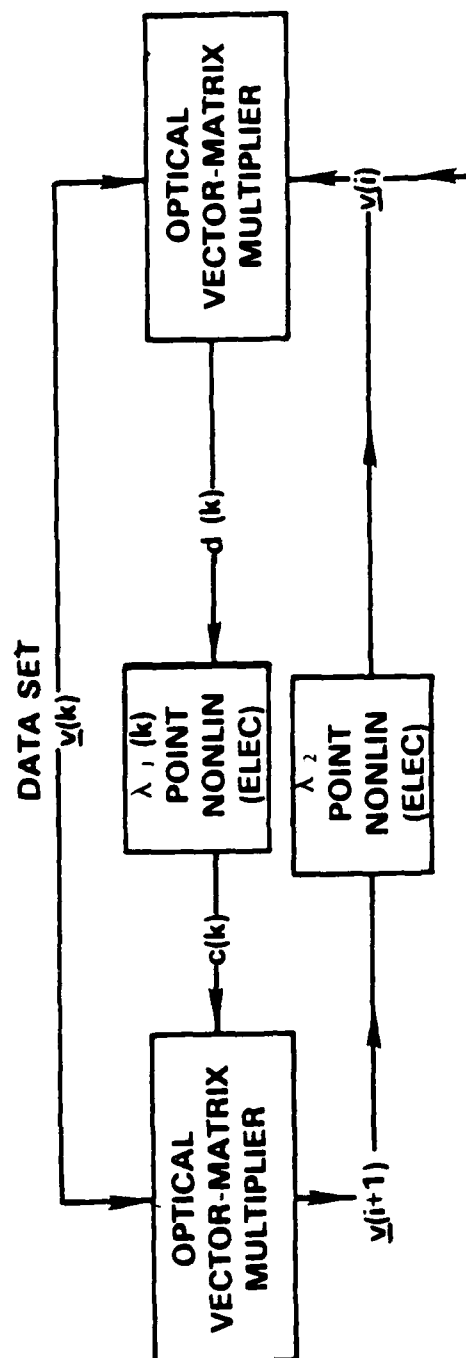
This basic model of associative memory can be modified in numerous ways, some of them discussed in Reference 4. In Reference 5 Hopfield suggests an iterative procedure where the estimate of the retrieved vector ( $\underline{v}$  in Equation (1)) is fed back to calculate an improved estimate. In that work, the data vectors were chosen to be binary, and this knowledge was used in hardclipping the retrieved vector before feeding it back in to

the system. Reference 1 discusses the optical implementation of the Hopfield model via an optical vector-matrix multiplier with feedback and a threshold nonlinearity in the feedback loop. The same procedure can be applied to the attentive associative memory model described in Equation (3) to improve the quality of retrieval.

In Reference 1 an extension of the Hopfield model to storage of images was proposed. Since images are 2-D matrices, their outer products will give us a 4-D tensor (corresponding to the recording step in Equation (1)). To facilitate the realization of this tensor, Reference 3 proposed an optical system that is equivalent to Equation (3) in that it also performs the inner product between the images first before forming the linear superposition of the stored images. The system, however, did not involve the nonlinear step contained in Equation (3). A recent paper by Soffer, et al,<sup>6</sup> discusses a holographic implementation of the associative memory for storing images, in which the correlation between the input and the stored images was subject to a nonlinear operation. That system did not contain provision for a channel dependant nonlinearity and therefore for attention.

### C. IMPLEMENTATION OF ATTENTIVE ASSOCIATIVE MEMORY

The block diagram of an attentive associative memory with iterative retrieval is shown in Figure (1). It consists of two vector-matrix multipliers (VMM) that are connected in a loop with nonlinear processing steps between both of them. The first VMM projects the input vector onto the space spanned by the stored vector. After the projection values (or the correlations) are nonlinearly processed, a reverse projection is performed bringing the vector back from projection space to the data space. This new vector is now subject to a nonlinearity that reflects our a priori knowledge about the nature of the data vector. This processed vector is again projected on the space spanned by the data vectors and the procedure is repeated till a stable state is reached.



#### HARDWARE REQUIREMENTS:

1-D SLM'S (OPTICALLY OR ELECTRONICALLY ADDRESSED)  
N-ELEMENTS

2-D SLM'S (OPTICALLY OR ELECTRONICALLY ADDRESSED)  
N X M ELEMENTS

1-D ELECTRONIC NONLINEAR ELEMENTS AND DETECTORS  
N-ELEMENTS

Figure 1. The block diagram of an optical attentive associative memory.



a. Computer Simulations

The performance of this simplest model of attentive associative memory was tested on a personal computer. A 4 X 4 binary image was converted into a 16-element binary vector and used as the basic datum. A set of four images was chosen for storage. Figure (2) shows the four binary images along with their auto- and cross-correlations. It can be seen that these images are highly correlated with the auto-to-cross correlation ratio of only 2. The cross-correlations between all possible pairs were also found to be uniform and equal to 4. This rather large cross-talk implies that the conventional associative memory models will have difficulties during retrieval, since the simple model shown in Equation (1) works well with orthogonal or near orthogonal set of vectors only. Knowing the level of cross-talk inherent in the data allows us to choose the appropriate nonlinear transformation (i) shown in Figure (3). It contains an offset (chosen to be 4 to suppress cross-talk), a linear portion, and saturation to a value of 1 when the correlation exceeds 8. In effect we have forced orthogonality on the data set by incorporating the nonlinearity shown in Figure (3). The a priori knowledge of binary nature of stored vectors is utilized by hardclipping the retrieved vector before feeding it back into the processor.

The results of a computer simulation are shown in Figure (4). The conventional associative memory failed to work even when a noise-free vector was available for retrieval. On the other hand, it was noted that as many as three bits out of the nonzero bits of the stored data could be missing and the attentive associative memory will still reconstruct the correct data vector. The hardclipping in addition to the threshold nonlinearity in the correlation domain gave a rapid convergence with the correct vector being retrieved in a single step.

Since the threshold nonlinearity essentially created an orthogonal set of vectors, another scheme of orthogonalizing the vectors was studied. The vectors in Figure (2) were converted into bipolar vectors by replacing all 0's with -1's. The resultant set of four bipolar images are shown in Figure 5(a). Since these images are orthogonal, we do get a

# STORING FOUR 16-ELEMENT VECTORS AND RETRIEVING THEM:

1	1	1	1	1	0	1	0	1	0	0	1	1	0	0	1
0	0	0	0	1	0	1	0	1	0	1	1	0	1	1	0
1	1	1	1	1	0	1	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	1	0	1	0	1	0	0	1	0	1

## CORRELATIONS:

8	4	4	4	4
4	8	4	4	4
4	4	8	4	4
4	4	4	8	8

Figure 2. Four binary images to be stored and their mutual inner products (correlations).

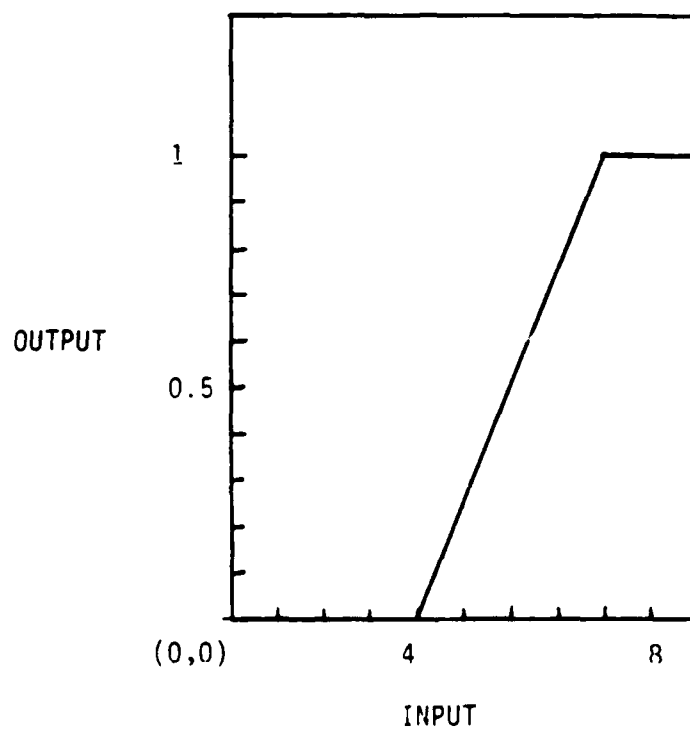


Figure 3. The nonlinear transfer function applied to the inner products (corresponds to  $1(k)$  in Figure 1).

**ASSOCIATIVE MEMORY RETRIEVAL STABLE STATES:  
THE OUTPUT IS CLIPPED AT 8 TO BINARIZE THE DATA -**

① 0 ① ①	① 0 ① 0	① 0 ① 0	① 0 ① ①
0 0 0 0	0 0 ① 0	0 ① 0 0	0 ① ① 0
① ① ① 0	① 0 ① 0	① 0 ① 0	① ① ① 0
0 0 0 0	① 0 0 0	0 0 0 ①	① 0 0 ①

**ATTENTIVE ASSOCIATIVE MEMORY STABLE STATES:  
THRESHOLD NONLINEARITY IN THE TRANSFORM PLANE:**

THRESHOLD SET AT 4

① 0 ① ①	① ① ① ①
0 0 0 0	0 0 0 0
① ① ① 0	① ① ① ①
0 0 0 0	0 0 0 0
INPUT	OUTPUT

**PERFECT RETRIEVAL WITH UP TO THREE MISSING BITS**

Figure 4. Performance of a conventional associative memory and an attentive associate memory in recalling the images shown in Figure 2.

perfect retrieval with the conventional associative described in Equation (1) when the complete input vector is presented. Figure 5(b) shows the results of the retrieval when the input contained three bits of error (-1 instead of +1). In the case of the first example error-free retrieval was achieved. The second example, however, shows that a different distribution of the errors will lead to an incorrect retrieval. This proves that with an orthogonal set of vectors, the cross-correlations can be significantly different for the same number of bit-errors. This is not the case with achieving orthogonality through a nonlinear step in the correlation domain.

b. Optical Implementation

Figure (1) indicated that the attentive associative memory contains two vector-matrix multipliers connected in a loop with nonlinearities in between them. The matrices in both the multipliers were however identical. This fact can be exploited to design an optical attentive associative memory with bi-directional propagation of light and a common matrix mask. A compact structure can be realized by using long finger-like modulators and detectors to perform the operation of broadcasting and summing, respectively, that are needed in vector-matrix multipliers. The schematic diagram of such a compact architecture is shown in Figure (6). The active part of the system consists of an optoelectronic panel containing pairs of detectors and light modulators that are electrical connected to each other through an amplifier and a nonlinear circuit. The current out of the detector stripe is proportional to the sum of the light distribution on it. That signal is then amplified and processed before applying it to the light modulator stripe, which then broadcasts it to its entire length uniformly. The system shown in Figure (6) is designed to store three vectors, each with four elements. Thus the input panel contains four pairs of detector-modulator stripes, each three-elements long. The other panel is placed in the correlation domain and contains three pairs of detector-modulator stripes, each four-elements long, that are orthogonally oriented with respect to the input panel stripes. The matrix mask sandwiched between the two panel contains the three four-element vectors.

A. ORTHOGONAL DATA VECTORS

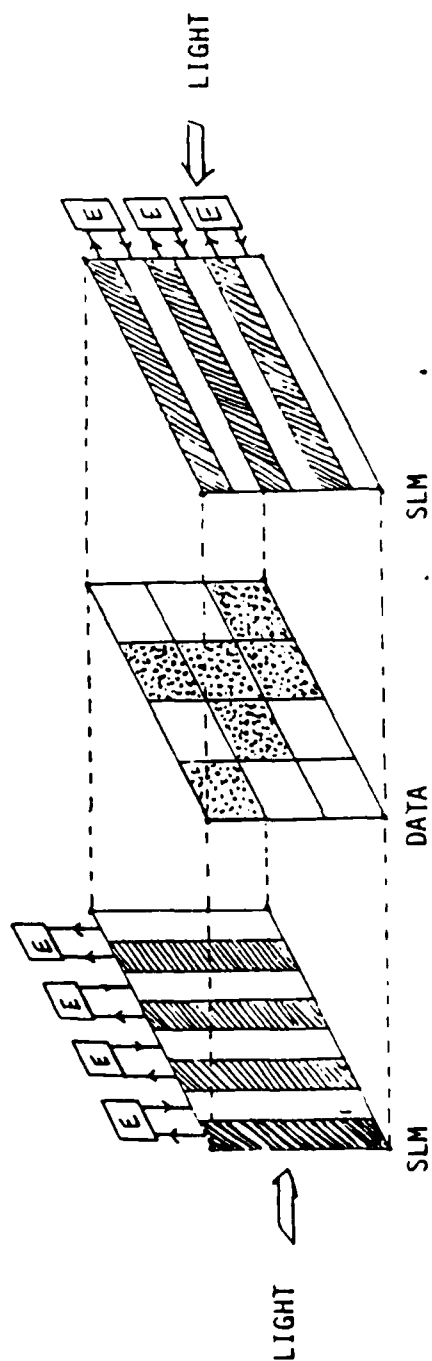
+	+	+	+	+	-	+	-	+	-	+	-	+	-	-	+
-	-	-	-	+	-	+	-	-	+	-	+	-	+	+	-
+	+	+	+	+	-	+	-	+	-	+	-	-	+	+	-
-	-	-	-	+	-	+	-	-	+	-	+	+	-	-	+

B. THREE ERRORS IN THE FIRST VECTOR

+	+	⊖	⊖		+	+	+	+
-	-	-	-		-	-	-	-
+	+	+	⊖	RETRIEVES	+	+	+	+
-	-	-	-		-	-	-	-
INPUT								
+	⊖	+	+		+	⊖	+	⊖
-	-	-	-		-	-	-	-
+	⊖	+	⊖	RETRIEVES	+	⊖	+	⊖
-	-	-	-		-	-	-	-

Figure 5. The retrieval characteristics of a conventional associative memory with orthogonal data set.

COMPACT OPTICAL ARCHITECTURE FOR AAM



- o SPECIALLY DESIGNED 1-D SLM'S (OPTICALLY ADDRESSED)
- o NO IMAGING/FOCUSING OPTICS
- o BI-DIRECTIONAL PROPAGATION OF LIGHT
- o CANDIDATE SLM'S:
  - PRIMO (HUGHES RESEARCH LABS)
  - Si/PLZT SLM (UC SAN DIEGO)

Figure 6. Compact optical architecture for an attentive associative memory.

The initial vector can be applied to the input panel via an optical signal to the photodetector or via an electrical signal to the modulator. This vector is then broadcast to all of the vectors of the matrix mask via the stripe modulator. The transmitted light contains the element-by-element multiplication between the initial vector and all the stored vectors. The stripe detector in the correlation domain now sums the products along a row thus performing a vector-vector inner product. These inner product (correlation) results are nonlinearly amplified and applied to the modulators, which broadcast them to the corresponding row vector in the matrix mask. The backward propagating light now performs a scalar-vector multiplication per channel. The detectors in the input pannel now perform a weighted sum of all the stored vectors, thus calculating a new estimate of the initial vector. The detector outputs are nonlinearly amplified before driving the modulator stripes, at which point the cycle repeats.

The system shown in Figure (6) can be simulated with discrete off-the-shelf components, such as LED's and photodetectors. The schematic diagrams of the input and the correlation plane panels consisting of LED's and photodetectors is shown in Figure (7). Three discrete photodetectors connected in parallel replace the stripe detector and three LED's connected in series replace the stripe modulator. An electronic amplifier module per channel implements the desired nonlinear amplification of the signal. An optoelectronic testbed capable of storing four 16-bit vector was fabricated. Figure (8) shows the photograph of the finished unit. The initial vectors can be input via 16 potentiometers on the front panel. The offset and gain in the correlation domain for each of the four stored vectors can also be controlled via 8 potentiometers on the front panel. One control adjusted the threshold level of the hardclipping operation in the input panel. A film mask was prepared encoding the vectors shown in Figure (2). The operation of this unit was tested and results consistent with the computer simulations were obtained.



# OPTO-ELECTRONIC TESTBED

THE BDM CORPORATION

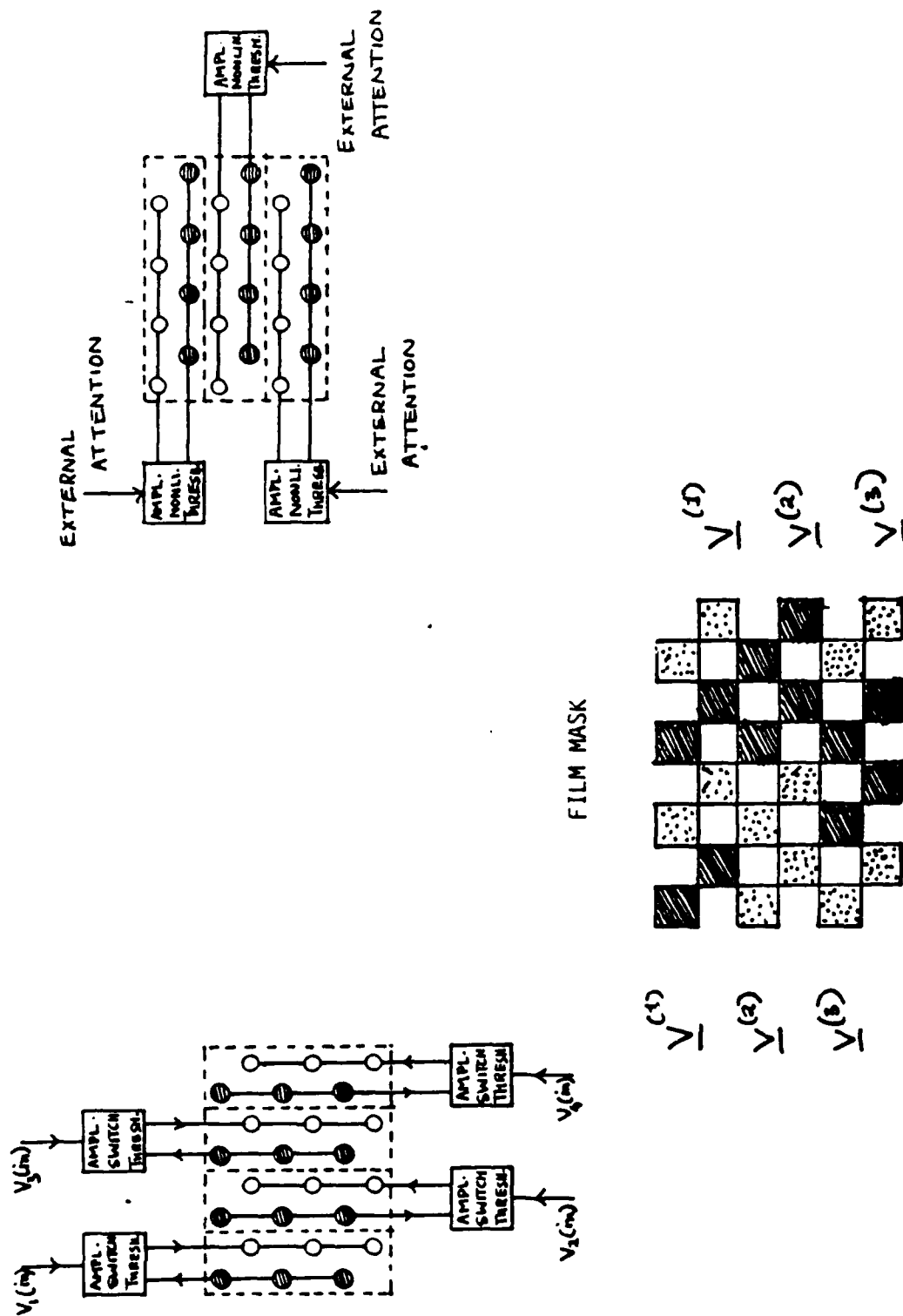


Figure 7. The schematic diagrams of the components of an optoelectronic testbed for realizing attentive associative memory.

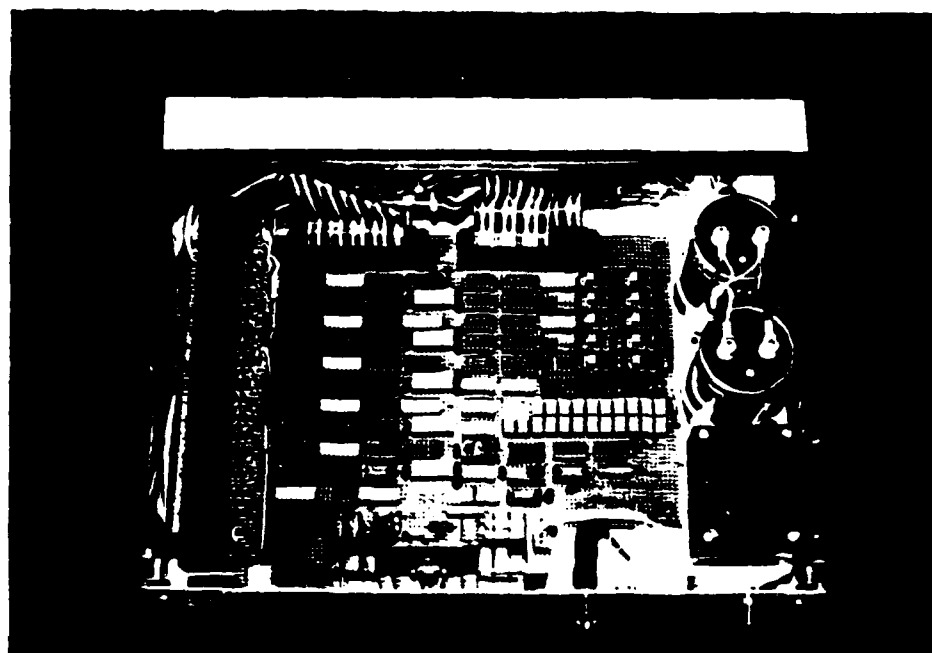
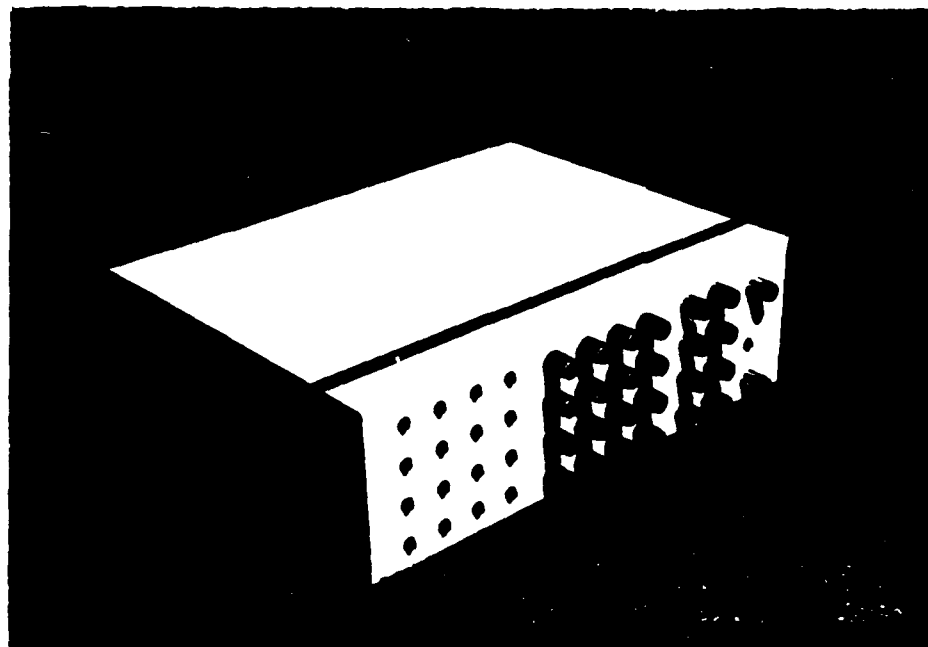


Figure 8. Photographs of the completed optoelectronic testbed and its interior.

D. FUTURE WORK

Further work on this project will proceed on several fronts. Replacing the hardclipping step with an offset-and-gain type of nonlinearity and making the offset adaptive is the first step that will be tried. The second most important step will be the study of the scaling issues involved in the optical implementation of the system shown in Figure (6) along with a study of candidate spatial light modulator designs. An extension of the attentive associative concept to a hetero-associative memory will be the next step. Making the stored vectors adapt in real time will require replacement of the film mask with a real time spatial light modulator. A study of different algorithms for incorporating learning in the attentive associative memory will be useful in determining the direction of that aspect.

E. ACKNOWLEDGEMENT

This research was supported by the Innovative Science and Technology Office of the Strategic Defence Initiative Organization and was administered through the Office of Naval Research under Contract number N00014-85-K-0479.

REFERENCES

1. D. Psaltis and N. Farhat, "Optical Information Processing Model Based on Associative Memory Models of Neural Nets with Thresholding and Feedback," Opt. Lett. Vol. 10, p. 98, 1985.
2. A. D. Fisher, C. L. Giles, and J. N. Lee, " Adaptive Associative Memory Models," Presented at Optical Computing Topical Meeting, 18-20 March, 1985, Lake Tahoe, Nevada.
3. Digest of The Annual Meeting of Optical Society, Oct. 20-24, 1985, Washington, DC.
4. T. Kohonen, "Self-organization and Associative Memory," Springer Series in Information Sciences, Vol. 8, Springer-Verlag Pub., 1984.
5. J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. Natl. Acad. Sci. USA 79, 2554, 1982.
6. B. H. Soffer, G. J. Dunning, Y. Owechko, E. Marom, "Associative Holographic Memory with Feedback using Phase Conjugate Mirrors," Submitted to Optics Lett.

**OPTICAL COMPUTING STRATEGIES**

**Richard Barakat, Principal Investigator**

**RGB ASSOCIATES, INC.**

RGB ASSOCIATES, INC.

RICHARD BARAKAT  
PRINCIPAL INVESTIGATOR

OPTICAL COMPUTING STRATEGIES

ABSTRACT

The research effort was devoted to two topics in optical computing. The first effort is of a very theoretical nature being an investigation into the inherent computational limits of incoherent optical computing systems in terms of a lower bound on the simultaneous volume and computing time resources of the system. This work is a generalization of the VLSI approach. The second effort is the development of a new algorithm for the multiplication of two rectangular matrices specifically designed to take full advantage of the fact that convolutions can be performed very rapidly using electro-optic and acousto-optic technology.

## TECHNICAL SUMMARY

### OBJECTIVES

1. Development of an algorithm for the multiplication of two rectangular matrices via convolution.
2. Development of a tractable mathematical model of an "optical" computing system (assuming incoherent light operations); use of such a model to investigate the inherent limits of computation in terms of a lower bound on the simultaneous resources of volume and computing time.

### DESCRIPTION OF WORK PERFORMED

Some of the research effort is written up as two independent reports (probably to be published):

1. An Algorithm for Matrix-Matrix  
Multiplication via Convolution.
2. Lower Bounds on the Computational  
Efficiency of Optical Computing Systems.

These reports are appended.

## CONCLUSIONS AND RECOMMENDATIONS

With respect to the matrix algorithm, we feel that it has the potential to be very useful. As such we recommend that it be implemented by some group with appropriate equipment. In order to conclude the theoretical aspects of the algorithm, an additional effort should be carried out to assess the numerical stability of the algorithm.

With respect to the computational efficiency problem, it does not seem worthwhile at this point in time to continue until more is known about device technology.



SECTION ONE

AN ALGORITHM FOR MATRIX-MATRIX  
MULTIPLICATION VIA CONVOLUTION

One of the virtues of electro-optical computing is the ability to carry out convolution operations very rapidly. Given this technical advantage, it is worthwhile to develop an algorithm for the multiplication of two rectangular matrices using convolution.

To this end let us consider the matrix product  $\hat{C} = \hat{A}\hat{B}$  where  $\hat{A}$  is of size  $n_1 \times n_2$ ,  $\hat{B}$  is of size  $n_2 \times n_3$ , and  $\hat{C}$  is of size  $n_1 \times n_3$ . Let the corresponding matrix elements be  $a_{ij}$ ,  $b_{jk}$ , and  $c_{ik}$ . Associate with  $\hat{A}$  and  $\hat{B}$  the polynomials  $P(x)$  and  $Q(x)$ , with  $x$  being interpreted as an indeterminate

$$P(x) = \sum_{s=0}^{(n_1-1)n_2n_3+n_2-1} p_s x^s \quad (1)$$

$$Q(x) = \sum_{t=0}^{n_2n_3-1} q_t x^t \quad (2)$$

Note that the degree of  $P(x)$  involves not only the size of  $\hat{A}$  through  $n_1$  and  $n_2$  but also the size of  $\hat{B}$  through  $n_3$ . The degree of  $Q(x)$ , on the other hand, involves only the size of  $\hat{B}$ , namely  $n_2$  and  $n_3$ . The  $p$  and  $q$  coefficients are related to the matrix elements of  $\hat{A}$  and  $\hat{B}$  by

$$p_s = a_{ij}, \quad \text{if } s = (i-1)n_2n_3 + j - 1 \quad (3a)$$

$$= 0, \quad \text{if } (i-1)n_2n_3 + n_2 \leq s \leq in_2n_3 \quad (3b)$$

and

$$q_t = b_{jk}, \quad \text{if } t = kn_2 - j \quad (4a)$$

$$= 0, \quad \text{if } t \geq n_2 n_3 \quad (4b)$$

with:  $1 \leq i \leq n_1$ ,  $1 \leq j \leq n_2$  and  $1 \leq k \leq n_3$ .

We claim that the elements of the matrix product  $\hat{C}$  are given by selected coefficients of the polynomial

$$\begin{aligned} R(x) &= P(x)Q(x) \\ &= \sum_{m=0}^{n_1 n_2 n_3 - 1} r_m x^m \end{aligned} \quad (5)$$

where

$$r_m = \sum_{s=0}^m p_s q_{m-s} \quad (6)$$

is the discrete convolution of the  $p$  and  $q$  coefficients. These selected  $r_m$  are given by

$$r_m = c_{ik}, \quad \text{if } m = (i-1)n_2 n_3 + kn_2 - 1. \quad (7)$$

A formal proof (which is really a verification of the formulae) is now given. We begin by rewriting Eq. (6) in the form

$$r_m = \sum_s p_s q_{m-s} = \sum_{\alpha, \beta, \gamma, \delta} a_{ij} b_{jk} \quad (8)$$

where the summation in the second series is over:

$$\alpha: \quad s = (i-1)n_2n_3 + j - 1 \quad (9a)$$

$$\beta: \quad (i-1)n_2n_3 < s < (i-1)n_2n_3 + n_2 \quad (9b)$$

$$\gamma: \quad t = m - s = kn_2 - j \quad (9c)$$

$$\delta: \quad t < n_2n_3 \quad (9d)$$

The  $\alpha$  term is simply Eq. (3a), while the  $\beta$  term is the negation of Eq. (3b). The  $\gamma$  term follows from Eq. (4a), while the  $\delta$  term is the negation of Eq. (4b). Upon substitution of the  $\alpha$  term into the  $\beta$  inequality, we immediately see that this can only be true

$$1 \leq j \leq n_2 \quad (10)$$

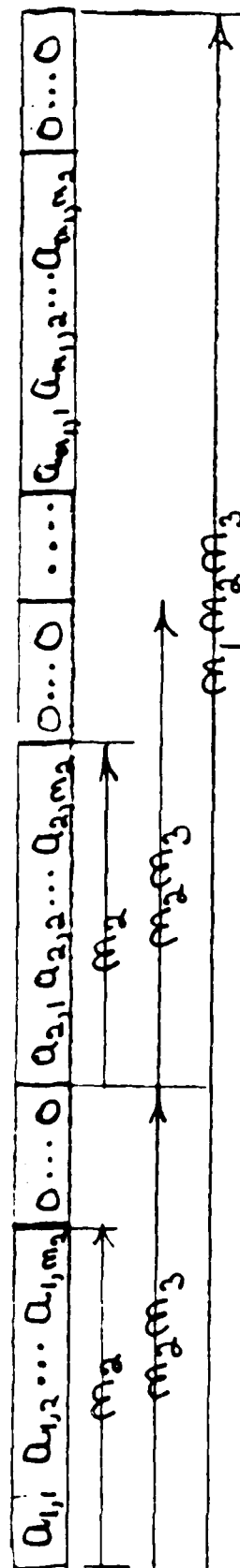
In like fashion, substitution of the  $\gamma$  term into the  $\delta$  inequality leads to the requirement that

$$m = (i-1)n_2n_3 + kn_2 - 1 \quad (11)$$

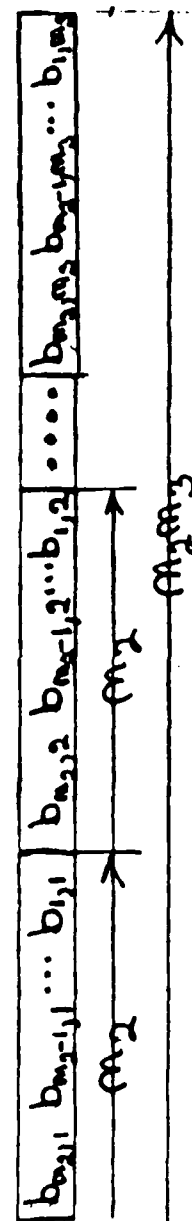
which is Eq. (7). Thus the formulae are verified.

A construction which leads to the various formulae for  $p_s$  and  $q_t$  in terms of  $a_{ij}$  and  $b_{jk}$  respectively uses row vectors. Consider a row vector  $\hat{p}$  whose elements we denote by  $p_s$  (coefficients of the polynomial  $P(x)$ ) composed of the matrix elements  $a_{ij}$  of  $\hat{A}$  and strings of zeros as depicted in Fig. 1A. The range of  $s$  is

$$0 \leq s \leq n_1n_2n_3 - n_2n_3 + n_2 - 1 \quad (12)$$



(A)



(B)

Fig. 1. Layout of the  $\hat{p}$  vector, see (A), and the  $\hat{q}$  vector, see (B).

consequently

$$p_s \equiv 0, \quad \text{if} \quad s \geq (n_1-1)n_2n_3 + n_2 \quad (13a)$$

$$\equiv 0, \quad \text{if} \quad s \leq n_2n_3. \quad (13b)$$

Furthermore the  $p_s$  are related to the  $a_{ij}$  as given by Eq. (3a), as the reader can verify by construction.

In like fashion, we construct another new row vector  $\hat{q}$  with elements  $q_t$  according to Fig. 1B. Unlike  $\hat{p}$ ,  $\hat{q}$  has no strings of zero elements. The range of  $t$  is

$$0 \leq t \leq n_2n_3 - 1 \quad (14)$$

so that

$$q_t = 0, \quad \text{if} \quad t \geq n_2n_3. \quad (15)$$

Within the range of  $t$ , the  $q_t$  are related to the  $b_{jk}$  by

$$q_t = b_{jk}, \quad \text{if} \quad t = (k-1)n_2 + n_2 - j \quad (16)$$

which reduces to Eq. (4a).

As an illustrative example of the algorithm, consider the case where  $A$  is  $2 \times 2$ ,  $B$  is  $2 \times 3$  so that  $\hat{C}$  is  $2 \times 3$  (i.e.,  $n_1 = 2$ ,  $n_2 = 2$ ,  $n_3 = 3$ ). The upper limits on the polynomials  $P$ ,  $Q$  and  $R$  are 7, 5, and 11, respectively. The  $p_s$ ,  $q_t$  and  $r_m$  coefficients evaluated according to Eqs. (3), (4) and (7) are listed in Table 1. Upon carrying out the convolution operation, Eq. (6), in conjunction with this table we have:

Table 1. Listing of the  $p$ ,  $q$  and  $t$  coefficients for the case where  $\hat{A}$  is  $2 \times 2$ ,  $\hat{B}$  is  $2 \times 3$  and  $\hat{C}$  is  $2 \times 3$ .

	$p_s$	$q_t$	$r_m$
0	$a_{11}$	$b_{21}$	
1	$a_{12}$	$b_{11}$	$c_{11}$
2	0	$b_{22}$	
3	0	$b_{12}$	$c_{12}$
4	0	$b_{23}$	
5	0	$b_{13}$	$c_{13}$
6	$a_{21}$		
7	$a_{22}$		$c_{21}$
8	0		
9	0		$c_{22}$
10	0		
11	0		$c_{23}$
12	0		

$$r_1 = c_{11} = p_0 q_1 + p_1 q_0 = a_{11} b_{11} + a_{12} b_{21} \quad (17a)$$

$$r_3 = c_{12} = p_0 q_3 + p_1 q_2 = a_{11} b_{12} + a_{12} b_{22} \quad (17b)$$

$$r_5 = c_{13} = p_0 q_5 + p_1 q_4 = a_{11} b_{13} + a_{12} b_{23} \quad (17c)$$

$$r_7 = c_{21} = p_6 q_1 + p_7 q_0 = a_{21} b_{11} + a_{22} b_{21} \quad (17d)$$

$$r_9 = c_{22} = p_6 q_3 + p_7 q_2 = a_{21} b_{12} + a_{22} b_{22} \quad (17e)$$

$$r_{11} = c_{23} = p_6 q_5 + p_7 q_4 = a_{21} b_{13} + a_{22} b_{23} \quad (17f)$$

These are, of course, the matrix elements as obtained by more standard procedures.

The implementation of the algorithm can be carried out in a straightforward fashion by re-examination of Figs. 1A and 1B. Note that the row vector  $\hat{p}$  in Fig. 1A consists of the *rows* of  $\hat{A}$  in which zeros are interspaced, the number of zeros is fixed. Thus we can easily handle the vector  $p$  containing the matrix elements  $a_{ij}$ . The row vector  $\hat{q}$ , containing the matrix elements  $b_{jk}$ , is simply the *columns* of  $\hat{B}$  in reverse order, see Fig. 1B. This vector is also easily handled in the implementation.



SECTION TWO

LOWER BOUNDS ON THE COMPUTATIONAL  
EFFICIENCY OF OPTICAL COMPUTING SYSTEMS

The advent of *Very Large Scale Integrated* (VLSI) circuitry has lead to considerable decrease in the physical size of computers with a corresponding increase in speed of execution of operations. Basically there are three interrelated aspects to VLSI: design and fabrication of the chips, design of systems which use these chips for specific applications, and development of algorithms which utilize the inherent capabilities of such chips. The revolution in computer science, for both numerical and nonnumerical applications, brought about by VLSI continues unabated.

The computational limitations of VLSI were first investigated by Thompson [1]. For an introduction to this work see the basic text of Ullman [2] which contains references to subsequent work. It has been shown that any VLSI circuit with area  $A$  and time  $T$  requires at least  $AT^2 = \Omega(n)$  to solve various computational problems such as FFT, convolution, and  $\ell \times \ell$  matrix multiplication where  $n = \ell^2$ . The symbol  $\Omega$  is defined in Ullman:  $f(n) = \Omega(g(n))$  means that there exists a positive constant  $c$  such that for an infinite number of values of  $n$  we have  $f(n) \geq cg(n)$ .

Nevertheless, VLSI suffers from the limitation that the technology upon which it relies is inherently two-dimensional. Snyder's recent review [3] contains a very useful discussion of the constraints imposed by VLSI as regards planarity. In particular conventional VLSI chips are constructed by superposing a small number of layers on top of a substrate. This substrate has a thickness which is order of magnitude greater than the size of the transistors and wire width. Input and output from a conventional VLSI chip must be made

by a limited number of pads located on the sides of the chip. VLSI chip technology is changing almost daily; however, some of the more basic aspects are discussed in Barbe [4] and Einspruch [5]. Although an ensemble of two-dimensional chips can be placed on top of each other with holes drilled down through them for interchip communication, the total number of layers is seriously limited by the substrate thickness of each chip: consequently the resulting device cannot properly be termed "three-dimensional VLSI". For this reason, it appears that truly three-dimensional VLSI will most likely not be possible to fabricate. Nevertheless some interesting theoretical investigations of three-dimensional VLSI have been carried out: Rosenberg [6], Leighton and Rosenberg [7].

The purpose of the present communication is to summarize investigations into various aspects of the computational performance of three-dimensional devices which make hybrid use of electronic and optical components to perform operations. Our goal is to facilitate general statements on such electro-optical computations with specific reference to lower bounds on their complexity. Since such devices may contain a large number of components, we term them VLSIO, with the O denoting optics.

We note that a very useful overview of optical computing (more properly electro-optical computing) may be found in Caulfield *et al.* [8].

In order to carry out such an analysis we outline the development of an abstract model of VLSIO which is essentially technology independent but incorporates the physical restrictions of light beam propagation as expounded by Gabor [9], especially with respect to the very important fact that the

amount of information passing through a cube of small volume is bounded. This physical constraint allows us to adapt previous VLSI lower bound arguments to the VLSIO situation and allows for comparisons of electro-optical computing devices in terms of their volume  $V$  and the time  $T$  taken by VLSIO on a given input ( $\equiv$  number of time units that elapse from the first input signal until the last output signal). We avoid making assumptions about the precise physics of the devices utilized. This would only limit the later application of these ideas as the physical models are improved and modified. Optical physics (through Gabor's theorem) implies an upper limit on the rate of information transfer across an optical beam, and hence a lower bound on computational efficiency of VLSIO. In addition we assume that any 2-D convolution of an  $n \times n$  array of points can be achieved by a VLSIO device in unit time step. This assumption is reasonable because there already exist optical devices which perform thusly.

Note that all the variables and functions are taken to be Boolean (i.e., the values of the variables are taken from  $\{0,1\}$ ).

We begin by discussing the well known abstract two-dimensional model of a VLSI chip as a  $L_1 \times L_2 \times L_3$  grid graph with height  $L_3$  ( $\ll L_1$  or  $L_2$ ) held constant. The distance between grid points is  $w$ , the feature width. The chip processors are located at various distance nodes of the grid graph with each processor storing a state consisting of  $b$  bits. Furthermore the processors execute synchronously on a step consisting of a time unit of duration  $t$  seconds. The remaining nodes are used for wire routing, or for input and output pods. Each wire can run along a path in the grid graph from

an input pod, or a processor, to various output pods, or processors. Wires are not allowed to intersect. On each time step, a value consisting of  $b$  bits of information is transmitted across the wire grid from either an input pod or a processor. The state of each processor is then updated on each step by a fixed function of the values transmitted by the wires leading into the processor, and by the state of the processor, in the previous step. The unit step transmission time across wires is justified by the fact that wire transmission can be made generally faster than transistor switching times. This remarkably simple model is sufficient to determine the computational efficiency of VLSI devices.

Following the two-dimensional version, the fundamental building block of our VLSIO device is the optical box  $B$ . It is a parallelopiped having lengths  $L_1$ ,  $L_2$  and  $L_3$  with input and output faces,  $F_{in}$  and  $F_{out}$ . These faces are assumed to take as input and as output two-dimensional integer arrays  $I(x,y)$  and  $O(x,y)$  respectively. For convenience, we consider the input sources and output detectors to be very small compared to the size of the optical box (in order to minimize optical diffraction effects), furthermore they are uniformly spaced a distance  $w$  apart. The input sources are taken to be LED's (laser emitting diodes) and the detectors are unspecified except to state that they are sensitive only to the *intensity* of the LED radiation. We remind the reader that most electro-optical computations are now performed via incoherent, geometrical optics based processors and not by coherent, Fourier transform based processors. The ancillary optical equipment (lenses, prisms, gratings, etc.) which spread and then collect the light can be

neglected in this version of the abstract model.

The output array is computed on each time step with a duration  $\tau$  as a fixed function  $\Delta_B$  of the input array;  $\Delta_B$  will, of course, depend upon the detailed optical characteristics of  $B$ .

The optical box, in addition to being three-dimensional, also differs from VLSI in another way; namely, optical beams rather than wires provide storage and cross-flow. Since the *modus operandi* is incoherent radiation, these beams can intersect without interacting. The basic question that now arises is: "to what extent do optical (laser) beams behave as wires?"

A wire can only transport information at a finite rate depending upon wire cross-section, skin effects, etc. We would also expect an optical beam to perform similarly notwithstanding the greater information rate. This problem has already been addressed by Gabor [9] who studied the "metrical information" in a light beam. The conclusion that he draws is that a light beam always has a *finite* upper limit with respect to information rates; the upper limit depending upon wavelength of light, smallest effective beam area, solid angle of divergence, etc. We need not concern ourselves with explicit formulae; for our purposes it suffices that we can interpret an optical beam as a wire.

Given this equivalence, we turn to the important problem of determining lower bounds (in terms of simultaneous volume and time) on the computational resources required for VLSIO to solve various problems.

In order not to unduly lengthen the text, it is assumed that the reader is familiar with Sections 1.4, 2.1 and 2.2 of Ullman's basic text [2].

Consider a Boolean function  $f$  with a set  $X$  of  $n$  input variables and a set  $Y$  of  $m$  output variables. Let  $X'$  be a subset of  $X$ ; also let  $P \equiv (X_L, X_R, Y_L, Y_R)$  where  $X_L, X_R$  and  $Y_L, Y_R$  are partitions of  $X$  and  $Y$  respectively. We term  $P$  balanced if between one-third and two-thirds of  $X'$  lies in  $X_L$  and note it by  $P_b$ . If  $\alpha$  and  $\beta$  are two input assignments, then we term them a fooling pair of assignments to  $X$  if:

- 1) output  $Y_L$  is distinct for input assignments  $\alpha(X)$  and  $\alpha(X_L)\beta(X_R)$
- 2) output  $Y_R$  is distinct for input assignments  $\beta(X)$  and  $\alpha(X_L)\beta(X_R)$ .

In addition, let the fooling set for  $P$  be a set of assignments  $A$  of  $X$  such that for all distinct  $\alpha, \beta \in A$ , at least one of  $(\alpha, \beta)$ ,  $(\beta, \alpha)$  is a fooling pair.

Finally, we require that the locations and times of the input and output are given only once.

Crucial to the analysis is the concept of information content (essentially "the amount of information that must cross a boundary in order to solve the problem"). Formally the information content of the Boolean function  $f$  is:

$$I_f = \max_{X'} \min_{P_b} \max_A \log_2(|A|) \quad (1)$$

where  $A$  denotes the fooling set corresponding to  $P_b$ . The following functions (of importance in electro-optical computing) are known to have information content  $I_f = \Omega(n)$ :

- a)  $n$  point discrete Fourier transforms.
- b) multiplication and inversion of two  $k \times k$  matrices where  $n = k^2$ .

c)  $n$  point convolution.

The following important result on lower bounds is due to Thompson [1,2]:  
Any two-dimensional VLSI chip computing a Boolean function  $f$  requires simultaneous area  $A$  and time  $T$  satisfying  $AT^2 = \Omega(I_f^2)$ .

We now prove: Any three-dimensional "optical box" computing a Boolean function  $f$  requires simultaneous volume  $V$  and time  $T$  satisfying  $VT^{3/2} = \Omega(I_f^{3/2})$ .

The proof (which we now sketch) is an adaptation of the two-dimensional technique. Let the device be a parallelepiped having dimensions  $L_1 \leq L_2 \leq L_3$  with volume  $V = L_1 L_2 L_3$ . Choose  $X'$  to be the subset of  $X$  such that  $I_f = I_f(X')$ . For each  $i = 1, 2, 3$  we can find a cut  $C_i$  of area

$$A_i \leq 2V/L_i \quad i = 1, 2, 3; \quad (2)$$

which disconnects the device into two components each of which contains at most two-thirds, but no less than one-third, of the inputs of  $X'$ . By definition at least  $I_f$  bits must be transported across each cut; this requires time

$$T \geq \frac{I_f}{A_i} \quad (3)$$

Consequently

$$V^2 T^3 \geq A_1 A_2 A_3 T^3 \geq I_f^3 \quad (4)$$

or

$$VT^{3/2} = \Omega(I_f^{3/2}) \quad (5)$$



which is the sought-for result. The main point to emphasize is that this result depends upon the fact that we can treat light beams as if they were wires.

An immediate consequence of this theorem is that the lower bounds for optical computing are:

- a)  $n$  point convolution or  $n$  point discrete Fourier transforms

$$VT^{3/2} = \Omega(n^{3/2}) \quad (6)$$

- b) multiplication and inversion of two  $\ell \times \ell$  matrices where  $n = \ell^2$

$$VT^{3/2} = \Omega(\ell^3) \quad (7)$$

These results follow from the statements quoted after Eq. (1). Equations (6) and (7) represent the lower bound performance of these two operations in terms of volume and time. It is important to remember that these bounds are a consequence of the fact that we allow the entire volume of  $B$  to be operative.

## REFERENCES

1. J.D. Ullman, *Computational Aspects of VLSI* (Computer Science Press, Rockville, Maryland, 1984), Chpt. 3.
2. C.D. Thompson, "A Complexity Theory for VLSI," Ph.D. thesis (Carnegie-Mellon Univ., Pittsburgh, 1981).
3. L. Snyder, "Supercomputers and VLSI: The effect of large scale integration in computer architecture,": in: *Advances in Computers*, vol. 25; M.C. Yovits, ed. (Academic Press, Orlando, Florida, 1984), pp. 1-33.
4. D.F. Barbe (editor), *Very Large Scale Integration (VLSI): Fundamentals and Applications* (Springer-Verlag, New York, 1980).
5. N.G. Einspruch (editor), *VLSI Electronics*, Microstructure Science, vols. 1-7 (Academic Press, Orlando, Florida, 1980-1985).
6. A.L. Rosenberg, "Three-dimensional integrated circuits," in: *VLSI Systems and Computation*; H. Kung, R. Sproul, and G. Steel, eds. (Computer Science Press, Rockville, Maryland, 1981), pp. 69-79.
7. F. Leighton and A. Rosenberg, "Three-dimensional circuit layouts," unpublished memorandum (MIT, Cambridge, 1982).
8. J. Caulfield, J. Neff, and W. Rhodes, "Optical computing: The coming revolution in optical signal processing," *Laser Focus/Electro-optics*, 38, 100-110 (1983).
9. D. Gabor, "Light and information," in: *Progress in Optics*, vol. 1; E. Wolf, ed. (North-Holland, Amsterdam, 1961), pp. 111-153.

**TIME-VARYING OPTICAL DATA PROCESSING FOR SDI**

David Casasent and B.V.K. Vijaya, Principal Investigators

CARNEGIE-MELLON UNIVERSITY  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**Carnegie-Mellon University**  
Department of Electrical and Computer Engineering  
Pittsburgh, Pennsylvania 15213

Principal Investigators:  
**David Casasent and B.V.K. Vijaya Kumar**

## **TIME-VARYING OPTICAL DATA PROCESSING FOR SDI**

### **Final Report for 1985**

*Prepared for:*  
ISTO Office of SDI  
Administered by ONR  
Purchase Order RI-39892

Steven C. Gustafson, Ph.D.  
University of Dayton  
Kettering Laboratories, 463  
Dayton, OH 45469

13 December 1985

### **ABSTRACT**

Our thrust is to apply optical data processing techniques to specific SDI problems, with attention to the presence of time-varying input data and the processing of more than one input frame of data. The specific applications defined in year one include: processing of time-varying imagery, detection and tracking of sub-pixel targets, optical Kalman filtering, and adaptive matched spatial filters. In each of these areas, new concepts, ideas, architectures, and algorithms suitable for optical data processing have been derived.

## TECHNICAL SUMMARY

### 1. OBJECTIVES

The objectives of this effort are:

1. to apply existing optical data processing engineering techniques to new SDI applications, and
2. to define and develop new optical data processing techniques for new SDF applications.

In area (1) above, we include optical matrix-vector processing. However, even with this new research area, there is a significant amount of basic research concerning algorithms, architectures, and number representations on which we have concentrated our work. In area (1), we also include optical correlation. New work in this area concerns new smart filters and adaptive filters. In area (2), we have addressed time-varying targets in which target detection decisions based upon multiple frames of data rather than one snapshot frame of data is required. We have also addressed the processing of sub-pixel target objects. These represent new research areas for optical data processing that are quite suitable for SDI applications.

## **2. SUMMARY OF WORK PERFORMED AND RESULTS OBTAINED**

We highlight our results below, including references to published, submitted and planned papers in the separate topical areas associated with each major task. All references that are not available presently in the open literature are included as appendices to this report. Brief summaries of these major topical areas now follow.

### **2.1 OPTICAL MATRIX-VECTOR PROCESSING**

We have addressed extended Kalman filtering applications in our matrix-vector studies. Our progress has been excellent in this area. This research area has been the task that has been given major emphasis in 1985. We now highlight the various sub-areas associated with this task.

#### **2.1.1 Architectures**

A laboratory prototype of a new basic matrix-vector optical architecture [1] using laser diode point modulators and a one-channel AO cell with frequency-multiplexing and multiple output detectors has been assembled. This architecture is quite versatile. It allows bipolar and complex-valued data handling. It also allows the use of analog processing or digital-encoded DMAC (digital multiplication by analog convolution) processing, as well as multi-level DMAC processing. These multiple processing modes are all achievable on the same processor. This is a quite unique feature for any

architecture (optical, or conventional digital multiprocessor). Tests and quantitative data on the performance of this system are expected in 1986. A second new architecture recently devised by us [2], uses a multi-channel AO cell, multiple input point modulators (achieved via an AO cell) and a linear detector CCD array (with a single-output channel). This architecture uses multi-channel AO cells to increase accuracy and processing capacity, but does not significantly increase the input/output and electronic support required. We have assembled the initial laboratory electronic support system for the first architecture [1] and are presently revising this system for our new second architecture. This is a major 1986 task item included in our proposed work. This is a quite significant effort in itself, but it is necessary and vital to obtain quantitative data and to demonstrate a real-time optical linear algebra laboratory system capable of reasonable problem solutions, rather than initial demonstrations only. We expect to learn much concerning future research direction from these initial lab tests.

### **2.1.2 Algorithms**

Parallel algorithms are necessary for parallel processors, such as our optical systems. We have devised a new LU decomposition algorithm suitable for our second architecture [2]. This algorithm is unique since it also marries the concepts of algorithms and architectures, i.e. the algotecture concept advanced earlier.

### **2.1.3 Number Representations**

This fertile research area refers to the fact that optical processors should not necessarily use conventional number representations employed in digital processors. Our work in this area has been most fruitful and has resulted in a unique negative base number representation [3] that we recently devised. This number representation is most suitable for optical processors using multi-channel and digital or multi-level DMAC algorithms for improved accuracy.

### **2.1.4 Case Studies**

The extended Kalman filter (EKF) was selected as our major case study to be detailed and studied. Our work in this area has been most productive [4,5] and fruitful during this brief six-month period. We have developed a new factorized EKF algorithm. We have married this algorithm with our new second optical linear algebra architecture and our new LU decomposition algorithm. We have detailed its data flow also. These features are lacking in most other algorithms. We have noted the increased efficiency and memory storage requirements that our new algorithm provides. Kalman filtering algorithms require floating point operations and performance in all instances. We have thus detailed how to achieve floating point accuracy on our optical processor [5]. Initial tests [4] (to be more fully addressed and detailed in 1986) indicate that this



factorized algorithm requires a significantly less accurate processor than does the conventional Kalman filtering algorithm. We intend to pursue this extensively in our 1986 research.

## **2.2 TIME-VARYING AND SUB-PIXEL TARGET DETECTION**

The optical processing of time-varying images and sub-pixel targets are two new optical data processing techniques to which initial attention has been given in our work and for which initial results have been obtained (October 1985, SDI presentation and associated viewgraph report). The concept we use in this processing involves modeling the background frame-to-frame as correlated noise with specified correlation lengths, mean and variances. The correlation of two such frames of background data thus results in a predicted correlation function shape. We employ exponential and parabolic models for these data correlations. From several (9 or 25) samples of the crosscorrelation of two successive frames of data, we can fit the samples obtained from this correlation plane to our models and estimate the sub-pixel shift between two frames of imagery. We then apply this sub-pixel shift and associated interpolation to the second image frame (this can be performed optically and is a 1986 task and new optical data processing technique and algorithm). We then difference the resultant two frames of data (this is also most suitable for optical implementation and is included in our 1986 proposed optical data processing research). The result is a frame with only moving

targets present. Full documentation of this algorithm and our initial results obtained with it is an additional 1986 task item. Extensions of our initial results are also most promising and are included as 1986 tasks. Our initial work and demonstrations represent the first application of optical data processing for time-varying data and sub-pixel target detection and the first and only successful demonstration (simulation) of such a concept.

### **2.3 ADAPTIVE MATCHED SPATIAL FILTERS**

Our architecture and algorithm for this concept have been defined and are summarized in a recent conference paper [6] and in the SDI Washington, D.C., October 1985 presentation (and its associated viewgraph report). The objective of this task is to update filters with new time-history input data. We achieve this with a computationally-efficient eigenvector algorithm. To update filters as required, we use a recursive algorithm. We have thusfar formulated this concept and selected several algorithms for use and for demonstrations. This represents a significant achievement. Details are expected in 1986. Our 1985 objective was primarily one of problem definition and problem formulation.

### **2.4 CONCLUSIONS AND RECOMMENDATIONS**

No definitive progress was made on our originally proposed nonlinear algorithms and neural processor work. This was due to the nature of the contract, in which

attention was given to educating students to provide such data, rather than employing faculty time to provide new innovative research ideas. In 1986, we propose to pursue these two areas to at least initial levels.

In 1985, we have extensively analyzed our new optical matrix-vector Kalman filter algorithm, architecture and number representation. All aspects of this work are proceeding quite well. Considerable basic research on architectures, algorithms, and number representation has evolved from this effort. This will be pursued further in 1986. Similar remarks apply to our sub-pixel and time-varying target cases. Initial new ideas on neural processors and revised approaches to conventional methods to address such processors have been considered (conceptually) and will be included as 1986 proposed research. Our major goal is to integrate optical image processing and optical linear algebra. This relates to the use of imaging and other sensors to initialize an optical Kalman filter for multi-target tracking applications.

#### REFERENCES

1. D. Casasent and J. Jackson, "Space and Frequency-Multiplexed Optical Linear Algebra Processors: Fabrication and Initial Tests", *Applied Optics*, Vol. 24,
2. D. Casasent and B.K. Taylor, "Banded-Matrix High-Performance Algorithm and Architecture", *Applied Optics*, Vol. 24, pp. 1476-1480, 15 May 1985.
3. C. Perlee and D. Casasent, "Negative-Base Encoding in Optical Linear Algebra Processors", *Applied Optics, Rapid Communications*, 15 January 1986.

4. J. Fisher, D. Casasent and C.P. Neuman, "A Factorized Extended Kalman Filter", *Proc. Soc. Photo Opt. Instr. Engrs.*, Vol. 564, August 1985.
5. J. Fisher, D. Casasent and C.P. Neuman, "A Factorized Extended Kalman Filter for Optical Processing", *Applied Optics*, submitted (see Appendix A).
6. D. Casasent, "Optical Computer Architectures for Pattern Analysis". *IEEE Computer Society, Workshop on Computer Architectures for Pattern Analysis and Image Database Management*, November 1985.

## APPENDIX A

### A FACTORIZED EXTENDED KALMAN FILTER FOR OPTICAL PROCESSING

James L. Fisher, David P. Casasent, Charles P. Neuman

Carnegie-Mellon University, Department of Electrical and Computer Engineering,  
Pittsburgh, PA 15213.

#### Abstract

Kalman filtering represents formidable linear algebra computational requirements for each new input measurement vector. An architecture-motivated implementation of a discrete-time extended Kalman filter (KF) algorithm is presented. This particular formulation takes advantage of the following features of the optical processor architecture: the ability to perform matrix-vector operations, floating-point capabilities, and specially designed matrix-vector  $\underline{L} \underline{U}$  decomposition operations. A factorized  $\underline{L} \underline{D} \underline{L}^T$  algorithm is used to propagate the covariance matrices between sample times. The air-to-air missile guidance problem is used as a case study wherein an extended Kalman filter (EKF) is required due to the nonlinear nature of the measurement equations.

#### Introduction

The pattern in the evolutionary cycle of architectural and algorithmic development is well established in the digital processor world, and we can now see a similar cycle forming in the optical processor world. Optical linear algebra processors (OLAP) [1] are attractive general-purpose systems for performing various linear algebra

matrix-vector operations. They are attractive by virtue of their parallelism, high-computational speeds, the ability to perform global operations, and their small size, weight, and power consumption. They are general-purpose systems that can solve a broad range of systems of linear algebra equations and perform a matrix-vector multiplication in a single step. Taking advantage of this capability, a novel method of performing  $\underline{L} \underline{U}$  decomposition was devised [2, 3]. This is an example of an architecture-motivated algorithm implementation.

In Section <sup>2</sup> 2, we review one OLAP architecture and discuss how it achieves bipolar and floating-point data capabilities. These data-handling methods will be seen to be an example of an algorithm-motivated architecture. In Section <sup>3</sup> 3, we review the fundamental KF and EKF requirements and establish our notation. The purpose of this paper is to provide an architecture-motivated implementation of the EKF for use in missile guidance applications. We seek an implementation of the EKF algorithm which makes optimum use of OLAP features, specifically matrix-vector intensive operations, and the use of the novel  $\underline{L} \underline{U}$  realization. The factorized EKF algorithm is advanced in Section <sup>4</sup> 4. All steps in the algorithm are detailed and therein the reader can see how it is designed with attention to the OLAP architecture.

Bierman and Thornton [4]- [7] provide the motivation for the study of factorized algorithms and single-precision EKF processors. Our application and algorithm extend this earlier work. In an EKF, the calculation of the covariance matrices are the major on-line operations required. Various formulations [4]- [9] have been advanced for this problem. Several digital [8, 9] and optical [10, 11] KF and EKF processors have been suggested and many OLAP architectures have been discussed [1]. However, only one paper [10] has detailed the realization of an EKF on an optical processor. Our present algorithm offers improved numerical stability and reduced computational accuracy requirements. The computations required in an EKF are of general use in other applications and thus these results should be of wide-spread use in areas besides highly-maneuverable target missile guidance with measurements that are nonlinear in Cartesian coordinates.

## 2.4 High-Accuracy Optical Linear Algebra Processors

The general-purpose OLAP used [3] is shown in Figure 1. It consists of a linear array of  $M$  point modulators stacked vertically at  $P_1$ . These are imaged vertically and expanded horizontally onto  $P_2$  which contains an  $N$ -channel acoustooptic (AO) cell. For discussion purposes, we consider  $M$  vertical regions of the AO cell at  $P_2$ . Each  $P_1$  point modulator uniformly illuminates one vertical region covering all  $N$  AO channels at  $P_2$ . Plane  $P_2$  is imaged horizontally and integrated vertically onto  $P_3$ . For simplicity, we

view  $P_3$  as a shift register linear detector array (the exact  $P_3$  system is more complex and is detailed elsewhere [3]). One vertical region originating at  $P_1$ , passing through the corresponding vertical region of  $P_2$ , and focused onto  $P_3$ , is called a *processor channel*; there are  $M$  processor channels in the system of Figure 1.

In Figure 1 we show a system with all  $M$  processor channel outputs summed at  $P_3$ . To achieve high-accuracy, this system implements a multiplication-by-convolution algorithm [12, 13, 14] which we now briefly review. In this algorithm, to multiply two numbers, their encoded bit streams are convolved (this yields a mixed-radix product) and the result is converted to the original encoded representation (by a simple shift/add procedure). To demonstrate, we multiply two binary-encoded numbers on the system of Figure 1, by considering a single-channel version of Figure 1, i.e.,  $M=1$ . The bits of the first number,  $s_1$ , are fed serially to  $P_1$  and the bits of the second number,  $s_2$ , are fed word-parallel to  $P_2$ . For  $N$ -bit words, a new bit enters  $P_1$  each  $T_1$  seconds and a new word enters  $P_2$  each  $T_2$  seconds, where  $T_2 = NT_1$ . The data incident on  $P_3$  each  $T_1$  are either  $s_2$  or zero (depending on the input bit at  $P_1$ ). The contents of  $P_3$  are shifted by one digit each  $T_1$  (our  $P_3$  detector system achieves this). At the next  $T_1$ , the incident data (again either  $s_2$  or zero) is added to the shifted output data produced at the prior  $T_1$ . One new mixed-binary output bit is thus produced each  $T_1$  (i.e., each



time the contents of  $P_3$  are shifted) and the full product is available after  $T_2$  seconds, and the 1-D output (in time) from  $P_3$  is the convolution of the bits of  $s_1$  and  $s_2$ , i.e., the mixed-radix product  $s_1 s_2$ . Thus, the  $P_3$  output  $s_{m3}(t)$  from the  $m$ -th channel is ~~the~~ the convolution (denoted by  $*$ ) of the data sequence  $s_{m1}(t)$  fed to the  $m$ -th point modulator at  $P_1$  and the data  $s_{m2}(x)$  present in the  $m$ -th region of  $P_2$ ; i.e.,

$$s_{m3}(kT_2) = s_{m1} * s_{m2} \quad (1)$$

where  $k$  is the discrete time index. This algorithm is quite versatile, since the signal sequences  $s_{m1}$  and  $s_{m2}$  can be multi-level or binary encoded representations of the two numbers [3]. In either case, the product in (1) is in mixed-radix representation. The outputs are easily converted to the original encoding representation by a single adder and A/D converter on the one  $P_3$  output line [3].

Next, we consider all  $M$  processor channels of the system and discuss how the system forms the vector inner product (VIP)  $\underline{a}^T \underline{b}$  of two  $M$ -element vectors. The  $M$  elements of vector  $\underline{a}$  are fed in parallel to  $P_1$  with each  $P_1$  point modulator fed serially (in time) with the encoded representation of each element as discussed before. Each vertical region of  $P_2$  (there are  $M$  such regions) contains the encoded representation of one element of the  $M$ -vector  $\underline{b}$ , with the  $N$  bits of that element placed horizontally across  $P_2$ , each bit in a different AO channel. Each processor channel performs a high-accuracy scalar product, and all  $M$  channels operate simultaneously. The output,  $P_3$ , is

the sum of these  $M$  different products formed on the  $M$  channels, or equivalently, the VIP  $\underline{a}^T \underline{b}$ . The number of bits of accuracy required in an application determines the number of AO channels  $N$  needed at  $P_2$ , the rate  $1/T_1$  at which encoded vector data need to be fed to  $P_1$ , and the specifications of the  $P_3$  adder and A/D converter. A reduced accuracy requirement can significantly increase speed (data input rate at  $P_1$ ) and reduce cost (the number of  $P_2$  channels). In our EKF application, the bit accuracy required as a function of the algorithm used will be considered. When multi-level encoding is used,  $N$  and  $1/T_1$  can be reduced (saving cost) or  $1/T_1$  can be increased (improving speed). The computation of one  $M$ -element VIP ( $M$  multiplications and  $(M-1)$  additions) accurate to 32 bits can be achieved every  $0.1 \mu\text{sec}$  using  $N=32$  channels (using binary encoding) or with only  $N=5$  channels (using base 4 encoding). With  $M=10$  (a modest number, and currently in use in our prototype), this system achieves about  $20 \text{ operations}/(0.1 \mu\text{sec}) = 200 \text{ MOPs}$ . The system in Figure 1 has other attractive features: It allows easy partitioning of a given problem and only one processor channel of the system is required to realize a unique implementation of  $\underline{L} \underline{U}$  decomposition [3].

The system also allows for the processing of floating-point data. In this case, the product mantissa is calculated optically and the exponent is handled by dedicated

external circuitry. Each vector element is expressed as a normalized mantissa and an exponent as in the conventional floating-point representation. For an M-element VIP, the largest of the M *individual* product exponents is found (its use is explained subsequently). Each multiplicand is then appropriately delayed (see below) and fed to the associated  $P_1$  channel.

To demonstrate these points, we consider a specific example. Let,

$$(a)_{10} = (a_3 \cdot a_2 a_1)_r \cdot r^{p_a} \text{ where}$$

- $(a)_{10}$  is the vector element in base 10 notation,
- $r$  is the base in the encoded representation of  $a$ ,
- $p_a$  is the exponent of the normalized base  $r$  representation of  $(a)_{10}$ ,
- $(a_3 \cdot a_2 a_1)_r$  is the base  $r$  normalized mantissa of  $(a)_{10}$ , and
- $a_i$  is a single base  $r$  bit of the normalized mantissa.

Note: by "normalized" we mean that  $a_3$  is the only non-zero bit to the left of the decimal point and  $a_3$  *must* be non-zero (except for  $(a)_{10}=0$ ). As our example, we

consider the binary representation (i.e.,  $r=2$ ) of five numbers:

$$\begin{aligned} (1.0)_{10} &= (1.00)_2 \cdot 2^{+0} \\ (1.5)_{10} &= (1.10)_2 \cdot 2^{+0} \\ (2.0)_{10} &= (1.00)_2 \cdot 2^{+1} \\ (2.5)_{10} &= (1.01)_2 \cdot 2^{+1} \\ (3.5)_{10} &= (1.11)_2 \cdot 2^{+1} \end{aligned}$$

We use these representations and consider the VIP

$$\begin{bmatrix} 3.5 & 1.0 & 2.5 \end{bmatrix} \begin{bmatrix} 2.0 & 1.5 & 1.0 \end{bmatrix}^T = 11.0.$$

By simple addition, the individual *product* exponents are found to be 2, 0, and 1, respectively, from which the largest exponent is  $p_{\max} = 2$ , corresponding to the product of 3.5 and 2.0. The mantissas of the second vector (the multipliers) are loaded directly into  $P_2$  (see Table 1). The mantissas of the first vector elements (the multiplicands) are loaded into  $P_1$  with the mantissa of vector element  $i$  delayed by  $\Delta_i T_1$  where  $\Delta_i$  is the difference between the largest product exponent,  $p_{\max}$ , and the  $i$ -th individual product exponent and  $\Delta_{i_{\max}}$  is the largest such difference. (see Tables 2).

The encoded and shifted mantissas are multiplied and summed, and the VIP result in mixed-radix form is obtained from  $P_3$  as a function of time as:

Output Time =	$7T_1$	$6T_1$	$5T_1$	$4T_1$	$3T_1$	$2T_1$	$1T_1$
Output Digit =	1	2	2	2	0	0	0

This mixed binary representation is converted to the normalized VIP mantissa value as

$$1 \cdot 2^0 + 2 \cdot 2^{-1} + 2 \cdot 2^{-2} + 2 \cdot 2^{-3} + 0 \cdot 2^{-4} + 0 \cdot 2^{-5} + 0 \cdot 2^{-6} = 2.75.$$

The normalized mantissa 2.75 is then multiplied by  $r^{**} \Delta_{i_{\max}} = 2^2$  to yield the VIP result 11.0. In this system, the time needed to compute a complete floating-point VIP product mantissa is increased from  $NT_1$  by  $\Delta_{i_{\max}} T_1$  (the time by which the multiplicand mantissa associated with the smallest individual product must be delayed). It should be noted that the digital realization of floating point operations requires a similar increase in time. The optical system realization has the advantages of parallelism, global operations, the use of non-binary encoding, etc.

### 3.4 Kalman Filter and Extended Kalman Filter Review

The objective of a KF and an EKF [15]- [18] is to estimate optimally (in the least-mean-squares sense) the state vector  $\underline{x}(t)$  of a dynamic system. We briefly detail the computational steps in a KF and an EKF in this section. Table 3 lists the notation and the dimensions of all quantities. The dynamics of the tracked system are modeled by:

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{C} \underline{u}(t) + \underline{w}_c(t). \quad (2)$$

The measurements  $\underline{z}(t)$  and state vector  $\underline{x}(t)$  are related by:

$$\underline{z}(t) = \underline{H} \underline{x}(t) + \underline{v}(t). \quad (3)$$

In all algebraic processors (optical or digital), new data are presented at discrete-time instants. We thus discretize the system dynamics model in (2) at the outset. By applying the standard exact discretization algorithm [16, 19, 20], the discrete-time system dynamics model and measurement model in (2) and (3) become:

$$\underline{x}_{k+1} = \underline{\Phi} \underline{x}_k + \underline{\Gamma} \underline{u}_k + \underline{w}_k \quad (4)$$

and

$$\underline{z}_k = \underline{H} \underline{x}_k + \underline{v}_k \quad (5)$$

where  $kT$  are the sampling times,  $T$  is the sampling period,  $\underline{\Phi} = \underline{\exp}(\underline{A}T)$ ,  $\underline{\Gamma} = \int_0^T \underline{\exp}(\underline{A}t) \underline{C} dt$ , and  $\underline{w}_k$  and  $\underline{v}_k$  are independent zero-mean white Gaussian noise vectors. The measurement noise  $\underline{v}_k$  is  $\underline{v}(t)$  evaluated at the sample time  $t=kT$ .

We refer the reader to Reference [16] for the relationship between  $\underline{w}(t)$  and  $\underline{w}_k$ .

The KF is an observer which takes into account the additive noise disturbances  $w_k$  and  $v_k$ , and produces the least-mean-square estimate  $\hat{x}_k$  of the state vector  $x_k$  from the new measurements  $z_k$ , and the *a priori* estimate  $\bar{x}_k$  calculated at time  $k-1$  (before the present measurement  $z_k$  was available). The discrete-time KF equations are listed in Table 4.

We notice that the second-order error covariance matrix calculations for  $P_k^{-1}$  and  $M_k$ , in (10) and (13), respectively, are independent of the state estimates  $\hat{x}_k$  and  $\bar{x}_k$ . Hence, from an initial error covariance matrix  $M_k$  or  $P_k$  at  $k=0$  and the noise statistics  $Q_k$  and  $R_k$  for all time  $k$ , we can *precompute* and store the KF gain matrix  $K_k$  for all time  $k$ . We need, therefore, to implement only (12) and (14) in real-time to compute  $\hat{x}_k$  and  $\bar{x}_k$ , and we never need to compute the second-order state estimate statistics  $M_k$  and  $P_k$  in real-time. This luxury of precomputability applies only for the case of *linear system and measurement models and known noise statistics*.

The EKF algorithm is compiled in Table 5. An EKF, rather than a KF, is used when either the system model or the measurement model is nonlinear. In a missile tracking application, the state vector  $x_k$  is in Cartesian coordinates whereas the measurement vector  $z_k$  is in polar coordinates and is thus nonlinear; i.e., the

measurements are bearing angles and range, and these are nonlinearly related to the Cartesian coordinates of position, velocity, and acceleration. The coordinate transformation vector  $\underline{h}[\underline{x}_k]$  in (15) and (18) represents this conversion. In an EKF, the linearized transformation matrix  $\underline{H}[\bar{\underline{x}}_k]$  in the  $\underline{P}_k^{-1}$  and  $\underline{K}_k$  equations (16) and (17) now depends on the *a priori* estimate  $\bar{\underline{x}}_k$ . We thus cannot precompute  $\underline{P}_k^{-1}$ ,  $\underline{K}_k$  and  $\underline{M}_{k+1}$ . Since we cannot precompute  $\underline{K}_k$ , all of the calculations in (13), (14), and (16)-(18) must be performed on-line during one sample period  $T$  for each new measurement. There is flexibility in the ordering of the operations in Table 5. For data flow and computation time efficiency, we implement Table 5 in the order (16), (17), (18), (14), and (13) for each time index  $k$ .

#### 4. Factorized EKF Algorithm

Bierman and Thornton [4]- [7] provided comparative studies on the numerical accuracy and efficiency of KF formulations using different factorized implementations. They found the factorized versions to be more numerically stable (i.e., the covariance matrices  $\underline{M}_k$  and  $\underline{P}_k$  always remained positive definite) and to require less computational precision. The basic concept in such an implementation is to factor  $\underline{M}_k$  and  $\underline{P}_k$  and to update only their factors at each time step  $k$ . Our factorized algorithm is unique in four ways:

1. We designed all steps to use matrix-vector operations rather than scalar operations.

2. We eliminate the need for most full matrix-matrix operations and thus simplify the processor and the algorithm to be implemented.
3. We implement an  $\underline{L} \underline{D} \underline{L}^T$  rather than an  $\underline{L} \underline{L}^T$  factorization or Householder factorizations as used by others [21, 22].  $\underline{L} \underline{L}^T$  factorizations were not used because they require a square-root operation which is not attractive. Householder factorizations were not employed because they are more complicated to achieve on a VIP processor. An  $\underline{L} \underline{D} \underline{L}^T$  factorization was selected because it is straightforward to implement using only one channel of the system in Figure 1. Bierman and Thornton apply the Given's algorithm, whereas our implementation of the  $\underline{L} \underline{U}$  decomposition algorithm [3], applied to the  $\underline{L} \underline{D} \underline{L}^T$  decomposition, operates on one row and column of the matrix in parallel, thus making more efficient use of our optical processor's parallel capabilities.
4. Bierman and Thornton compared the numerical accuracies of double-precision and single-precision computations. However, they computed all vector inner products using double-precision arithmetic and then rounded the results to single-precision, and they stored state estimates in double-precision. In contrast, we perform *all* computations using single-precision (floating-point) arithmetic.

In Table 6, we outline the algorithmic development and computational steps to obtain the factors of  $\underline{M}_{k+1}$  in (13). The method we use is to evaluate (13) as  $\underline{M}_{k+1} = \underline{\phi} \underline{P}_k \underline{\phi}^T + q_1 \underline{b}_1 \underline{b}_1^T + \dots + q_n \underline{b}_n \underline{b}_n^T$ , where  $q_i$  and  $\underline{b}_i$  are defined in Table 6.



This evaluation is performed recursively by adding one of the vector outer products (VOPs)  $q_i \underline{b}_i \underline{b}_i^T$  at a time as:

$$\underline{G}_{i+1} = \underline{G}_i + q_i \underline{b}_i \underline{b}_i^T \quad \text{for } i = 1, 2, \dots, n, \quad (6)$$

where  $i$  is the iteration index, and not the time index  $k$ , and  $\underline{G}_1 = \underline{\phi} \underline{P}_k \underline{\phi}^T$ . We seek the  $\underline{L} \underline{D} \underline{L}^T$  factors of  $\underline{M}_{k+1}$ , i.e., the factors of  $\underline{G}_{i+1}$  in (6) at  $i=n$ , and not the full  $\underline{M}_{k+1}$  or  $\underline{G}_{n+1}$  matrices. We thus write  $\underline{G}_i = \underline{L}_{G_i} \underline{D}_{G_i} \underline{L}_{G_i}^T$  and  $\underline{G}_{i+1} = \underline{L}_{G_{i+1}} \underline{D}_{G_{i+1}} \underline{L}_{G_{i+1}}^T$ . (The factors  $\underline{L}_G$  and  $\underline{D}_G$  are defined in Table 6.) The recursion in (6) can thus be written as

$$\underline{G}_{i+1} = \underline{G}_i + \underline{L}_{G_i} (q_i \underline{d}_i \underline{d}_i^T) \underline{L}_{G_i}^T = \underline{L}_{G_i} \underline{S}_i \underline{L}_{G_i}^T, \quad (7)$$

where  $\underline{d}_i$  is the solution of  $\underline{L}_{G_i} \underline{d}_i = \underline{b}_i$  and  $\underline{S}_i = \underline{D}_{G_i} + q_i \underline{d}_i \underline{d}_i^T = \underline{L}_{S_i} \underline{D}_{S_i} \underline{L}_{S_i}^T$ . We implement our algorithm by the recursive computation in (7). These steps are performed as follows:  $\underline{L}_{G_i}$  is available from the previous recursion  $i-1$ , and  $\underline{S}_i$  is computed at each recursion  $i$ . The  $\underline{L} \underline{D} \underline{L}^T$  factors of  $\underline{S}_i$  are then computed and the new  $\underline{L}_{G_{i+1}}$  and  $\underline{D}_{G_{i+1}}$  factors for  $\underline{G}_{i+1}$  are then easily obtained (updated) as  $\underline{L}_{G_{i+1}} = \underline{L}_{G_i} \underline{L}_{S_i}$  and  $\underline{D}_{G_{i+1}} = \underline{D}_{S_i}$ . Since each matrix in (7) is described by its  $\underline{L} \underline{D} \underline{L}^T$  factors, updating the factors of  $\underline{G}_i$  at each iteration is easily achieved.

We repeat (7) until  $i=n$  (the number of estimated states) and thus obtain the desired  $\underline{L} \underline{D} \underline{L}^T$  factors of  $\underline{M}_{k+1} = \underline{G}_{n+1}$  in (13). The programming for the  $\underline{L} \underline{D} \underline{L}^T$  algorithm to compute  $\underline{L}_M$  and  $\underline{D}_M$  for  $\underline{M}_{k+1}$  is presented in the Computation section

of Table 6. Steps 1-3 compute  $\underline{L}_{G_1}$  and  $\underline{D}_{G_1}$  for  $\underline{G}_1$  and are thus the initialization steps. Steps 5-10 compute  $\underline{L}_G$  and  $\underline{D}_G$  for one iteration  $i$ . These steps are repeated until  $i=n$  with  $\underline{L}_M$  and  $\underline{D}_M$  updated at each  $i$  as in Steps 9 and 10. All  $\underline{L}_G$  occupy the same physical memory location, and similarly for all  $\underline{D}_G$ . Our algorithm thus makes efficient use of memory storage. As shown, all operations, except those involving  $\underline{S}$ , use only diagonal and triangular matrices (thus simplifying the processor, algorithm, and data flow). The  $\underline{S}$  matrix is full, but computation of its  $\underline{L}_S$  and  $\underline{D}_S$  factors is quite simple using a Cholesky decomposition modification of our  $\underline{L} \underline{U}$  algorithm [2, 3] which is especially easy to realize on only one channel of the system in Figure 1. The algorithm's steps, operations, data flow and memory requirements are thus most attractive.

In order to calculate the *a priori* covariance  $\underline{M}_{k+1}$  described above, we need the factors  $\underline{L}_P$  and  $\underline{D}_P$  of the *a posteriori* covariance  $\underline{P}_k$ . We now consider that portion of the EKF algorithm which computes these  $\underline{L}_P$  and  $\underline{D}_P$  factors. Table 7 specifies our notation, algorithmic development, and the on-line update program. These steps parallel those in Table 6. Equation (16) gives an expression for  $\underline{P}_k^{-1}$ , whereas we desire instead the factors of  $\underline{P}_k$ . Thus, we apply the matrix inversion lemma of (20) (see Table 7) and obtain the recursive formula in (21) to compute  $\underline{P}_k$  by summing  $r$  VOPs,

where  $r$  is the number of measurements. As before, this requires that we process only one VOP at a time. The recursions are performed as in (22), with the new  $\underline{L}_G$  and  $\underline{D}_G$  computed as in (23). After  $r$  iterations, the factors  $\underline{L}_P$  and  $\underline{D}_P$  (which we desire) of  $\underline{P}_k$  are obtained.

The calculation of the  $\underline{P}_k$  factors (Table 7) and the  $\underline{M}_{k+1}$  factors (Table 6) are the most computationally intensive steps in the EKF algorithm (Table 5). Once the  $\underline{P}_k$  factors have been obtained, we compute the state estimate measurement-update (18) by first substituting the expression for the gain matrix (17) into (18) and performing the computations in the resultant equation from right to left (thereby requiring only matrix-vector operations). Computation of (14) is straightforward. We then compute (13) as in Table 6. This completes all of the EKF steps of Table 5. To summarize, these equations are implemented in the order of: a form of (16), a combination of (17) and (18), (14), and (13).

The key computational steps in Tables 6 and 7, are vector and not scalar operations. Such detail, formulation and development are required for data flow analysis and for evaluation of the computation time required. These aspects are unique in this work where we consider both the algorithm and the processor architecture to be

used for implementation. We note that  $(\Phi \underline{D})$ , and all  $\underline{L}$  matrices are unit diagonal lower triangular matrices, and  $\underline{D}$  and all  $\underline{D}$  matrices are diagonal. All operations (except Steps 7 and 8 in Table 6, and Steps 8 and 9 in Table 7) involve only triangular and diagonal matrices and are thus easily achieved with high accuracy and efficient data flow. The  $\underline{L}_S$  and  $\underline{D}_S$  decompositions required (Step 8 in Table 6 and Step 9 in Table 7) involve the full matrix  $\underline{S}$ , but are easily calculated [2, 3] using only one channel of the system in Figure 1. The matrices  $\underline{D}$ ,  $(\underline{D}^{-1})^2$ ,  $\Phi$ ,  $\Gamma$ ,  $\underline{L}_Q$ ,  $\underline{D}_Q$ , and  $\underline{R}$  are precomputed off-line once and stored in ROM. The data flow and computational sequence are such that all  $\underline{L}_G$ ,  $\underline{L}_P$ , and  $\underline{L}_M$  matrices can occupy the same physical storage location (and similarly for all  $\underline{D}_G$ ,  $\underline{D}_P$ , and  $\underline{D}_M$  matrices). The operations required in each time step  $k$  in the entire EKF can be performed in 1 msec on the VIP processor of Figure 1. In a subsequent publication, we will detail the performance of this EKF algorithm for several case studies.

## 5 Summary and Conclusions

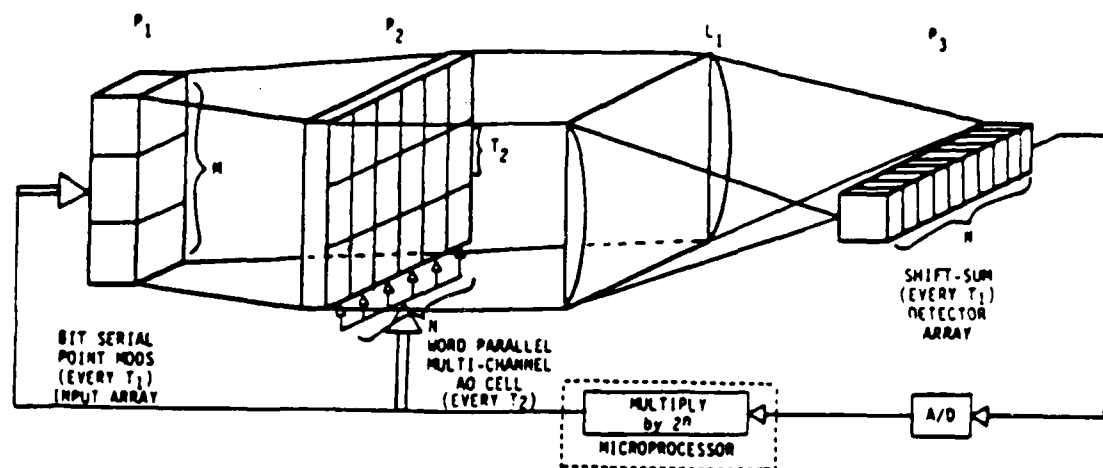
A high-accuracy floating-point optical linear algebra processor has been discussed and its use in EKF processing detailed. A new factorized EKF algorithm was advanced and detailed. All steps in this algorithm were formulated as vector intensive operations, with attention to data flow, the hardware processor and the algorithm. All steps required in one cycle of an EKF can be performed in 1 msec on the system described, thus allowing a new measurement every msec.

## Acknowledgments

The support of this research by grants from the Air Force Office of Scientific Research (AFOSR-84-0293) and by the Strategic Defense Initiative Office of Innovative Science and Technology through a UDRI subcontract is gratefully acknowledged.

## List of Figures

- Figure 1:** Simplified Schematic of a High-Accuracy Vector Inner Product Processor (N-bit Accuracy, M-element Vector)<sup>3</sup>. 17



**Figure 1:** Simplified Schematic of a High-Accuracy Vector Inner Product Processor (N-bit Accuracy, M-element Vector) [3].

## List of Tables

<b>Table 1:</b>	Timed Inputs to the AO Cell Channels at $P_2$	18
<b>Table 2:</b>	Timed Inputs to Point Modulators at $P_1$	19
<b>Table 3:</b>	Definition of Symbols Used in Discrete-Time Linear and Extended Kalman Filter Algorithms	20
<b>Table 4:</b>	System Model and Covariance Kalman Filter Algorithm	21
<b>Table 5:</b>	System Model and Covariance Extended Kalman Filter Algorithm	22
<b>Table 6:</b>	Computation of $\underline{M}_{k+1}$	23
<b>Table 7:</b>	Computation of $\underline{P}_k$	24



**Table 1:** Timed Inputs to the AO Cell Channels at  $P_2$

<u>Time</u>	<u>Multiplier</u>	<u>AO Channel Contents</u>		
		$a_3$	$a_2$	$a_1$
$1T_2$	3.5	1	1	1
$2T_2$	1.0	1	0	0
$3T_2$	2.5	1	0	1

**Table 2: Timed Inputs to Point Modulators at  $P_1$**

Point Modulator	Multiplicand Mantissa	Delay $\Delta_i$	Input Time Slots				
			$5T_1$	$4T_1$	$3T_1$	$2T_1$	$1T_1$
top	2.0	2	1	0	0	$0_d$	$0_d$
middle	1.5	0	$0_p$	$0_p$	1	1	0
bottom	1.0	1	$0_p$	1	0	0	$0_d$

where  $0_d$  are delay zeros and  $0_p$  are padding zeros

**Table 3: Definition of Symbols Used in Discrete-Time Linear and Extended Kalman Filter Algorithms**

<u>Symbol</u>	<u>Dimension</u>	<u>Description</u>
$E\{\cdot\}$		expectation operator
$\underline{\Gamma}$	$(n \times m)$	input distribution matrix
$\underline{h}[\underline{\bar{x}}_k]$	$(r \times 1)$	nonlinear coordinate transformation vector which transforms the Cartesian coordinates of $\underline{\bar{x}}_k$ to the polar coordinate frame of the measurement vector $\underline{z}_k$
$\underline{H}$	$(r \times n)$	linear coordinate transformation matrix (in KF)
$\underline{H}[\underline{\bar{x}}_k]$	$(r \times n)$	linearised measurement matrix (in EKF): $\underline{H}[\underline{\bar{x}}_k] = \partial \underline{h}[\underline{x}_k] / \partial \underline{x}_k$ evaluated at $\underline{x}_k = \underline{\bar{x}}_k$
$k$		discrete-time index
$\underline{K}_k$	$(n \times r)$	Kalman gain matrix
$\underline{M}_k$	$(n \times n)$	a priori error covariance matrix
$m$		number of inputs
$n$		number of states
$\underline{P}_k$	$(n \times n)$	a posteriori error covariance matrix
$\underline{\Phi}$	$(n \times n)$	system matrix
$\underline{Q}_k$	$(n \times n)$	system driving noise covariance matrix of $\underline{w}_k$
$r$		number of measurements
$\underline{R}_k$	$(r \times r)$	measurement noise covariance matrix of $\underline{v}_k$
$T$		data sampling period
$\underline{u}_k$	$(m \times 1)$	input (control) vector
$\underline{v}_k$	$(r \times 1)$	measurement noise vector
$\underline{w}_k$	$(n \times 1)$	system noise vector
$\underline{x}_k$	$(n \times 1)$	system state vector
$\underline{\hat{x}}_k$	$(n \times 1)$	a priori state vector estimate
$\underline{\hat{x}}_k$	$(n \times 1)$	a posteriori state vector estimate
$\underline{z}_k$	$(r \times 1)$	measurement vector

**Table 4: System Model and Covariance Kalman Filter Algorithm**

Linear System Model	$\underline{x}_{k+1} = \underline{\Phi} \underline{x}_k + \underline{\Gamma} \underline{u}_k + \underline{w}_k$	(8)
Linear Measurement Model	$\underline{z}_k = \underline{H} \underline{x}_k + \underline{v}_k$	(9)
Error Covariance Measurement-Update	$\underline{P}_k^{-1} = \underline{M}_k^{-1} + \underline{H}^T \underline{R}_k^{-1} \underline{H}$	(10)
Gain Matrix	$\underline{K}_k = \underline{P}_k \underline{H}^T \underline{R}_k^{-1}$	(11)
State Estimate Measurement-Update	$\hat{\underline{x}}_k = \bar{\underline{x}}_k + \underline{K}_k \{ \underline{z}_k - \underline{H} \bar{\underline{x}}_k \}$	(12)
Error Covariance Time-Update	$\underline{M}_{k+1} = \underline{\Phi} \underline{P}_k \underline{\Phi}^T + \underline{Q}_k$	(13)
State Estimate Time-Update	$\bar{\underline{x}}_{k+1} = \underline{\Phi} \hat{\underline{x}}_k + \underline{\Gamma} \underline{u}_k$	(14)

**Table 5: System Model and Covariance Extended Kalman Filter Algorithm**

Linear System Model	$\underline{x}_{k+1} = \underline{\phi} \underline{x}_k + \underline{\Gamma} \underline{u}_k + \underline{w}_k$	(8)
Nonlinear Measurement Model	$\underline{z}_k = \underline{h}[\underline{x}_k] + \underline{v}_k$	(15)
Error Covariance Measurement-Update	$\underline{P}_k^{-1} = \underline{M}_k^{-1} + \underline{H}^T[\underline{\bar{x}}_k] \underline{R}_k^{-1} \underline{H}[\underline{\bar{x}}_k]$	(16)
Gain Matrix	$\underline{K}_k = \underline{P}_k \underline{H}^T[\underline{\bar{x}}_k] \underline{R}_k^{-1}$	(17)
State Estimate Measurement-Update	$\hat{\underline{x}}_k = \underline{\bar{x}}_k + \underline{K}_k \{ \underline{z}_k - \underline{h}[\underline{\bar{x}}_k] \}$	(18)
Error Covariance Time-Update	$\underline{M}_{k+1} = \underline{\phi} \underline{P}_k \underline{\phi}^T + \underline{Q}_k$	(13)
State Estimate Time-Update	$\underline{\bar{x}}_{k+1} = \underline{\phi} \hat{\underline{x}}_k + \underline{\Gamma} \underline{u}_k$	(14)

Table 6: Computation of  $\underline{M}_{k+1}$

Development

- $\underline{M}_{k+1} = \underline{\phi} \underline{P}_k \underline{\phi}^T + \underline{Q} = \underline{\phi} \underline{P}_k \underline{\phi}^T + \underline{L}_Q \underline{D}_Q \underline{L}_Q^T$  (13)
- Column-wise partition  $\underline{L}_Q$  as  $\underline{L}_Q = [\underline{b}_1 \dots \underline{b}_n]$  where  $\underline{b}_i$  is an  $n$ -vector.
- Write  $\underline{D}_Q$  as  $\underline{D}_Q = \text{diag}[q_1 \dots q_n]$  where each  $q_i$  is a scalar.
- $\underline{M}_{k+1} = \underline{\phi} \underline{P}_k \underline{\phi}^T + q_1 \underline{b}_1 \underline{b}_1^T + \dots + q_n \underline{b}_n \underline{b}_n^T$ .
- Let  $\underline{G} = \underline{\phi} \underline{P}_k \underline{\phi}^T = (\underline{\phi} \underline{L}_P \underline{D})(\underline{D}^{-1} \underline{D}_P \underline{D}^{-T})(\underline{\phi} \underline{L}_P \underline{D})^T = \underline{L}_{G_1} \underline{D}_{G_1} \underline{L}_{G_1}^T$ , where  $\underline{D}$  is a diagonal matrix whose non-zero elements are the reciprocals of the corresponding diagonal elements of  $\underline{\phi}$ .
- In recursive form,  $\underline{G}_{i+1} = \underline{G}_i + q_i \underline{b}_i \underline{b}_i^T = \underline{L}_{G_i} \underline{S}_i \underline{L}_{G_i}^T$ . We first form  $\underline{S}_i = \underline{D}_{G_i} + q_i \underline{d}_i \underline{d}_i^T$  (where  $\underline{L}_{G_i} \underline{d}_i = \underline{b}_i$ ), and then decompose  $\underline{S}_i$  as  $\underline{S}_i = \underline{L}_{S_i} \underline{D}_{S_i} \underline{L}_{S_i}^T$ . Thus,  $\underline{L}_{G_{i+1}} = \underline{L}_{G_i} \underline{L}_{S_i}$  and  $\underline{D}_{G_{i+1}} = \underline{D}_{S_i}$ .
- The following are unit diagonal lower triangular matrices:  $\underline{L}_Q$ ,  $(\underline{\phi} \underline{D})(\underline{\phi}$  by itself is lower triangular but not unit diagonal),  $\underline{L}_P$ , Tmpl (in Step 1 below),  $\underline{L}_S$ , and  $\underline{L}_M$ .
- The following are diagonal matrices:  $\underline{D}_Q$ ,  $\underline{D}$ ,  $\underline{D}_G$ ,  $\underline{D}_P$ ,  $\underline{D}_S$ , and  $\underline{D}_M$ .
- The following are full matrices:  $\underline{S}$ , Tmpl (in Steps 6 and 7 below).

Computation

1. Tmpl  $\leftarrow \underline{\phi} \underline{L}_P$
2.  $\underline{L}_G \leftarrow \underline{Tmpl} \underline{D}$
3.  $\underline{D}_G \leftarrow (\underline{D}^{-1})^2 \underline{D}_P$
4.  $i \leftarrow 1$
5. Back substitute  $\underline{L}_G \underline{d}_i = \underline{b}_i$  for  $\underline{d}_i$ .
6. Tmpl  $\leftarrow \underline{L}_G \underline{d}_i \underline{d}_i^T$
7.  $\underline{S} \leftarrow \underline{D}_G + \underline{Tmpl}$
8. Compute the Cholesky factors  $\underline{L}_S$  and  $\underline{D}_S$  of  $\underline{S}$ .
9.  $\underline{D}_G \leftarrow \underline{D}_S$
10.  $\underline{L}_G \leftarrow \underline{L}_G \underline{L}_S$
11. If  $i=n$  then  $\underline{L}_M \leftarrow \underline{L}_G$  and  $\underline{D}_M \leftarrow \underline{D}_G$ ; else  $i \leftarrow i + 1$  and go to Step 5.
12. Stop.

Table 7: Computation of  $\underline{P}_k$

Development

- Column-wise partition  $\underline{H}^T [ \bar{\underline{X}}_k ]$  as  $[ \underline{h}_1 \dots \underline{h}_r ]$  where  $\underline{h}_i$  is an  $n$ -vector.
- Let  $\underline{R}_k^{-1} = \text{diag}[ \underline{R}_1^{-1} \dots \underline{R}_r^{-1} ]$ .
- Rewrite (16) as  $\underline{P}_k^{-1} = \underline{M}_k^{-1} + \underline{R}_1^{-1} \underline{h}_1 \underline{h}_1^T + \dots + \underline{R}_r^{-1} \underline{h}_r \underline{h}_r^T$
- We will process one vector outer product at a time, therefore we use the notation

$$\underline{G}_{i+1}^{-1} = \underline{G}_i^{-1} + \underline{R}_i^{-1} \underline{h}_i \underline{h}_i^T \quad (19)$$

where  $\underline{G}_i^{-1} = \underline{M}_k^{-1}$  and  $\underline{P}_k^{-1} = \underline{G}_{r+1}^{-1}$ . (Subscripts  $k$  and  $i$  are time and iteration indices, respectively.)

- Take the inverse of both sides of (19) and apply the matrix inversion lemma

$$(\underline{A} + \underline{X} \underline{C} \underline{Y}^T)^{-1} = \underline{A}^{-1} - \underline{A}^{-1} \underline{X} (\underline{C}^{-1} + \underline{Y}^T \underline{A}^{-1} \underline{X})^{-1} \underline{Y}^T \underline{A}^{-1} \quad (20)$$

with the substitutions  $\underline{A} = \underline{G}_i^{-1}$ ,  $\underline{X} = \underline{Y} = \underline{h}_i$ , and  $\underline{C} = \underline{R}_i^{-1}$  to obtain

$$\underline{G}_{i+1} = \underline{G}_i + \alpha (\underline{G}_i \underline{h}_i) (\underline{G}_i \underline{h}_i)^T \quad (21)$$

where the scalar  $\alpha$  satisfies  $1/\alpha = -(\underline{R}_i + \underline{h}_i^T \underline{G}_i \underline{h}_i)$ .

- Let  $\underline{G}_i = \underline{L}_G \underline{D}_G \underline{L}_G^T$  and  $\underline{d}_i = \underline{D}_G \underline{L}_G^T \underline{h}_i$ . Then (21) becomes

$$\underline{L}_{G_{i+1}} \underline{D}_{G_{i+1}} \underline{L}_{G_{i+1}}^T = \underline{L}_{G_i} [ \underline{D}_{G_i} + \alpha \underline{d}_i \underline{d}_i^T ] \underline{L}_{G_i}^T = \underline{L}_{G_i} \underline{S}_i \underline{L}_{G_i}^T \quad (22)$$

If  $\underline{S}_i = \underline{L}_{S_i} \underline{D}_{S_i} \underline{L}_{S_i}^T$  is the factored form of the matrix  $\underline{S}_i = \underline{D}_{G_i} + \alpha \underline{d}_i \underline{d}_i^T$ , then

$$\underline{L}_{G_{i+1}} = \underline{L}_{G_i} \underline{L}_{S_i} \text{ and } \underline{D}_{G_{i+1}} = \underline{D}_{S_i} \quad (23)$$

Computation

1.  $i \leftarrow 1$
2.  $\underline{L}_G \leftarrow \underline{L}_M$  and  $\underline{D}_G \leftarrow \underline{D}_M$
3.  $\underline{\text{Tmp1}} \leftarrow \underline{L}_G^T \underline{h}_i$
4.  $\underline{d}_i \leftarrow \underline{D}_G \underline{\text{Tmp1}} = \underline{D} \underline{L}_G^T \underline{h}_i$
5.  $\underline{\text{Tmp1}} \leftarrow (\underline{\text{Tmp1}})^T \underline{d}_i$
6.  $\alpha \leftarrow -1/(\underline{\text{Tmp1}} + \underline{R}_i)$
7.  $\underline{\text{Tmp1}} \leftarrow \alpha \underline{d}_i \underline{d}_i^T$
8.  $\underline{S} \leftarrow \underline{D}_G + \underline{\text{Tmp1}}$
9. Compute the Cholesky factors  $\underline{L}_S$  and  $\underline{D}_S$  of  $\underline{S}$ .
10.  $\underline{D}_G \leftarrow \underline{D}_S$  ( $\underline{D}_G$  is the appropriate component of the left side of (21) at Step  $i+1$ )
11.  $\underline{L}_G \leftarrow \underline{L}_G \underline{L}_S$
12. If  $i=r$ , then  $\underline{L}_P \leftarrow \underline{L}_G$  and  $\underline{D}_P \leftarrow \underline{D}_G$ ; else  $i \leftarrow i+1$  and go to Step 3.
13. Stop.

## References

1. "Special Issue on Optical Computing", *Proceedings of the IEEE*, Vol. 72, July 1984.
2. D. Casasent and A. Ghosh, "LU and Cholesky Decomposition on an Optical Systolic Array Processor", *Optics Communications*, Vol. 46, July 1983, pp. 270-273.
3. D. Casasent and B. K. Taylor, "Banded-Matrix High-Performance Algorithm and Architecture", *Applied Optics*, Vol. 24, 15 May 1985, pp. 1476-1480.
4. G. J. Bierman, "A Comparison of Discrete Linear Filtering Algorithms", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-9, January 1973, pp. 28-37.
5. G. J. Bierman and C. L. Thornton, "Numerical Comparison of Kalman Filter Algorithms: Orbit Determination Case Study", *Automatica*, Vol. 13, January 1977, pp. 23-35.
6. G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York, 1977.
7. C. L. Thornton and G. J. Bierman, "UDU<sup>T</sup> Covariance Factorization for Kalman Filtering", in *Control and Dynamic Systems*, C. T. Leondes, ed., Academic Press, New York, Vol. 16, 1980, pp. 117-248.
8. R. H. Travassos, "Real-Time Implementation of Systolic Kalman Filters", *Proceedings of the SPIE*, Vol. 431, August 1983.
9. "Special Issue on Applications of Kalman Filtering", *IEEE Transactions on Automatic Control*, Vol. AC-28, March, 1983.
10. D. Casasent, C. P. Neuman, and J. Lycas, "Optical Kalman Filtering for Missile Guidance", *Applied Optics*, Vol. 23, 15 June 1984, pp. 1960-1966.
11. J. R. Jackson and D. Casasent, "State Estimation Kalman Filter Using Optical Processing: Noise Statistics Known", *Applied Optics*, Vol. 23, 1 February 1984, pp. 376-378.
12. E. Swartzlander, "The Quasi-Serial Multiplier", *IEEE Transactions on Computing*, Vol. C-22, April 1973, pp. 317-321.
13. H. J. Whitehouse and J. M. Speiser, "Linear Signal Processing Architectures", in *Aspects of Signal Processing with Emphasis on Underwater Acoustics, Part 2*, G. Tacconi, ed., D. Reidel, Boston, 1976, pp. 669-702.
14. D. Psaltis, D. Casasent, D. Neft, and M. Carlotto, "Accurate Numerical Computation by Optical Convolution", *Proceedings of the SPIE*, Vol. 232, April 1980, pp. 151-156.



15. A. Gelb, editor, *Applied Optimal Estimation*, The MIT Press, Cambridge, MA, 1974.
16. R. A. Singer, "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-6, July 1970, pp. 473-483.
17. Peter S. Maybeck, *Stochastic Models, Estimation, and Control*, Academic Press, New York, Vol. 1, 1979.
18. Peter S. Maybeck, *Stochastic Models, Estimation, and Control*, Academic Press, New York, Vol. 2, 1982.
19. G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, Addison-Wesley, Reading, MA, 1980.
20. J. J. D'Azzo and C. H. Houpis, *Linear Control System Analysis and Design*, McGraw-Hill, New York, 1981.
21. C. L. Thornton, "Triangular Covariance Factorizations for Kalman Filtering", Technical Report 33-798, Jet Propulsion Laboratory, Pasadena, CA, 1976.
22. B. Friedlander, T. Kailath, M. Morf, and L. Ljung, "Extended Levinson and Chandrasekhar equations for general discrete-time linear estimation problems", *IEEE Transactions on Automatic Control*, Vol. AC-23, No. 4, August 1978, pp. 653-659.

**NEW METHODS FOR OPTICAL COMPUTING**

H. J. Caulfield, Principal Investigator  
Joseph Shamir  
Mostafa Abushagur  
Stephen Welsted

THE CENTER FOR APPLIED OPTICS  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING  
MATHEMATICS AND STATISTICS DEPARTMENT  
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

## NEW METHODS FOR OPTICAL COMPUTING

University of Dayton Research Institute  
Order Number RI 39887

### ABSTRACT

This report will cover three independent developments in the field of computing. Two of these developments are aimed at achieving extremely high accuracy results with only moderately accurate components. This appears to be a new direction in optical computing. The third approach is also a total new direction in optical computing. It involves a new type of optical device called an Optical Fredkin Gate. Optical Fredkin gates appear to have virtually an unlimited number of functions. Among those logical operations, memory, and interconnection. They have the additional virtue of being, in principle, capable of extremely fast switching. We will devote one section each to those three developments

by

H. J. Caulfield  
Joseph Shamir  
Mostafa Abushagur  
Stephen Welsted

Final Report

For the Period 8 June, 1985 - 31 December, 1985

This research was supported by  
The Center for Applied Optics  
Department of Electrical and Computer Engineering  
Mathematics and Statistics Department  
The University of Alabama in Huntsville  
Huntsville, Alabama 35899

31 December 1985

## I. TECHNICAL SUMMARY

### 1. OBJECTIVES

The purpose of this work was to devise new approaches to optical computing that could have the affect of creating new opportunities for the use of optics in SDI. Three new approaches were investigated and all appear worthy of additional study. First, we investigated the possibility of doing numerical algebra to high accuracy with relatively low accuracy but fast and inexpensive analog optical processors. This required bootstrapping by doing a few relatively simple operations with electronic digital processors. Both theory and simulation were studied. Convergence to highly solutions appears to be possible in a great many cases. Second, we thought of ways using relatively low accuracy optical technologies to do a variety of neighborhood processors and, in particular cellular array processors, that is rather arbitrary and programmable non linear operations on images were studied. The methods devised appear to be able to work at extremely high speeds and over extremely large neighborhoods. Third, we devised a totally new type of optical computing component-the Optical Fredkin Gate. Optical Fredkin Gates appear to be capable of performing any logical operations, performing any delay or memory operation on a sequence of bits, or doing any interconnect. They are, therefore, perhaps the most general optical computer component devised. In some configurations, optical fredkin Gates have the potentiality of operating at the source bandwidth with limit ( $10^{12} - 12^{14}H$ ).

### 2. Description of Work Performed

#### (a) Optical Fredkin Gates

One of the limitations imposed on increasing computation power, be it electronic or optic, stems from the large amount of energy that needs to be dissipated during computer operation<sup>1</sup>. Part of this energy is due to the intrinsic

sic nature of the traditional logic elements. This fact becomes evident if we recall that conventional logic gate has more input lines than output lines. Thus some of the information coming into the gate is lost and cannot be retrieved. The irreversible nature of the gate makes it dissipative not only in information but also in energy. In an effort to overcome these limitations, Fredkin<sup>2</sup> proposed a new kind of logic gate which has the same number of output lines as it has input lines. Fredkin gates are capable of performing conventional logic operations while preserving all the original information. In contrast to the conventional logic gates the fredkin gates may, in principle, be run backwards to regenerate the original input signals.

The purpose of this work is to introduce the Optical Fredkin Gate which may become one of the basic building blocks of an optical computer. An overview of the main aspects of the Fredkin Gate is given in the next section, followed by a variety of proposed optical implementations. A number of useful applications are discussed in a final section.

Background of the Fredkin Gates: The basic Fredkin Gate is defined as a black box having three binary inputs and three binary outputs (Fig. 1). the C-input - the control line, determines the operations of the gate on the other two inputs according the following rules:

$$\begin{array}{lll} \text{IF} & C = 0: & A' = A; \quad B' = B; \\ \text{IF} & C = 1: & A' = B; \quad B' = A; \end{array} \quad (1)$$

It is quite evident that this gate is reversible, i. e. it may be run backward to return to the original inputs and therefore it is in principle non-

detector (Photoconductor or photodiode-amplifier combination). Polarizing beam-splitters may be applied whenever a spatial separation is required between the A and b lines. The main advantage of this gate is its relative simplicity while its disadvantage is the different nature of the C-line that also changes level during transition through a gate (i.e. there is a lower light intensity in C' than i c. This effect may, however, be corrected by incorporating an amplifying medium on the line).

In figure 4 we show a schematic diagram of the acousto-optic gate: The two input lines are laser beams incident on an acousto-optic deflector (either bulk or integrated SAW) at the bragg angle. If there is no acoustic signal ( $C = 0$ ), the two beams continue unaffected (A' and B') while if C is present each beam is deflected into the other channel. This is also a imple gate but, here too, one has a C line which is basically different in nature than the other two lines. Nevertheless this kind of gate can be easily cascaded and intergrated. For example, a single acoustic pulse may activate many gates as it travels along the system.

The photorefractive gate, based on four-wave-mixing is an all optical gate with one of its tentative implementations illustrated in Fig. 5. In this case the C-line constitutes the two pump beams. The inputs A and b are transmitted if C is absent and phase-conjugated when the pump is present resulting in a switching between the outputs.

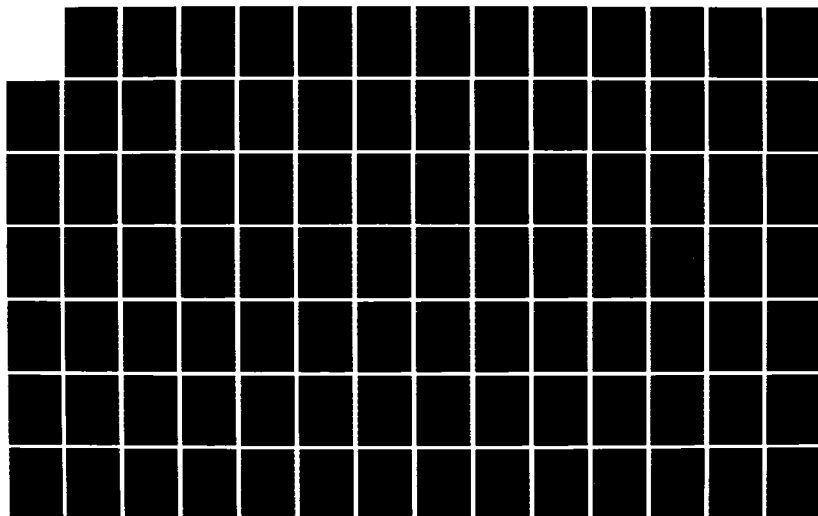
In optical communication and integrated optical systems a modulated waveguied or fiber coupler may serve as a Fredkin gate. Two general classes of this kind of gates may be implemented. The out-of-plane control, shown schematically in Fig. 6a, and in the inplane control with one possibility depicted in Fig. 6b. A number of workers have already implemented the electronically

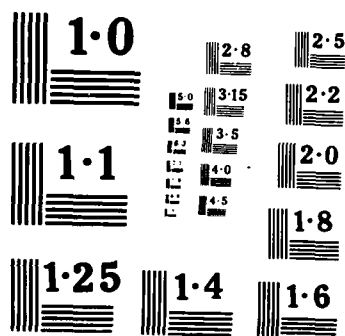
AD-A169 882 OPTICAL PROCESSING(U) DAYTON UNIV OH RESEARCH INST 2/6  
S C GUSTAFSON 31 DEC 85 UDR-TR-85-148 N00014-85-K-0479

UNCLASSIFIED

F/G 9/2

NL







addressed coupler<sup>4,5</sup> that may serve as a Fredkin Gate with an electronic C-input. To symetrize the system one may use photodetection combined with the electro-optic coupler to facilitate optical control. A more advanced technology would be the use of photorefractive material for direct optical control of the coupling constant. The example in (b) is a wave guide coupler incorporating highly anisotropic guides containing nonlinear material. The two coupling waves (A and B) are introduced with the same polarization so that they can couple while the control signal, C, is orthogonally polarized in such a way that its power is used to activate the coupling between the A and b channels but it does not couple itself into the other guide<sup>6</sup>.

Proposed Devices Incorporating Fredkin Gates: We demonstrate the applicability of these new gates by proposing, in addition to the conventional logic gates, two very useful devices that incorporate arrays of the waveguide gates shown in Fig. 6.

The Optical Crossbar. The gate array of Fig. 7 may be constructed of gates of the type depicted in Fig. 6a or the type of Fig. 6b. In the first case each gate may be accessed randomly from above by an electric field or by light, depending on the specific device used. As we are dealing with optical computing we might prefer activation by light such as a holographic coupler<sup>8</sup> or fiber coupler. With proper addressing each input line can be coupled to each output line. This system may prove to be an extremely fast and efficient crossbar or optical switchboard. The in-plane addressing of Fig. 6b is applicable if one desires to activate a whole column together. At first sight it appears that this kind of addressing is not suitable for random access; however with very fast pulses this also becomes feasible.

The Tapped Delay Line: The basic configuration of Fig. 8a is a tapped delay line. A fiber ring may be utilized for long delays while for very

short delays one may use waveguide rings the feasibility of which has also been demonstrated<sup>9,10</sup>. Here too, the addressing may be of the first type (Fig. 6a) or of the second type (Fig. 6b). Such a setup may be used to delay all the energy in a pulse or just part of it to produce a pulse train from a single initial pulse. A slight modification of the system as illustrated in Fig. 8b may be used to reverse the direction of signal flow resulting in a true reversible Fredkin gate. In the future, an optical memory block may resemble the array depicted in Fig. 8c. This seems to be a short term memory, but with the integration of amplifying medium it may serve also as a long-term memory.

Discussion: Conventional approaches to optical computing followed the lines put forward by workers with electronic systems. Traditional logic gates are well suited for electronic computing but may not be the best choice for optical processors. In this work we indicated that one should also consider different implementations for optical computing systems with one very promising possibility being the Fredkin gate. These gates have many simple optical implementations and may prove to be very fast and energy efficient. The various implementations and applications given here are just samples to indicate the diverse possibilities available.

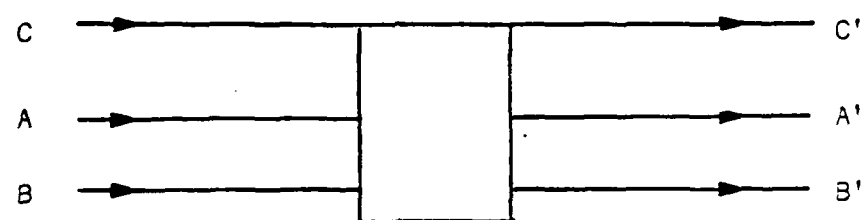


Fig.1

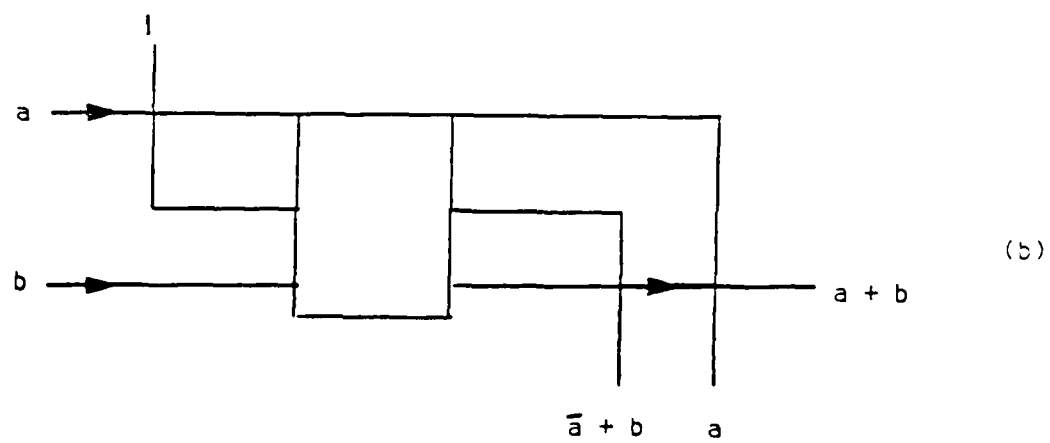
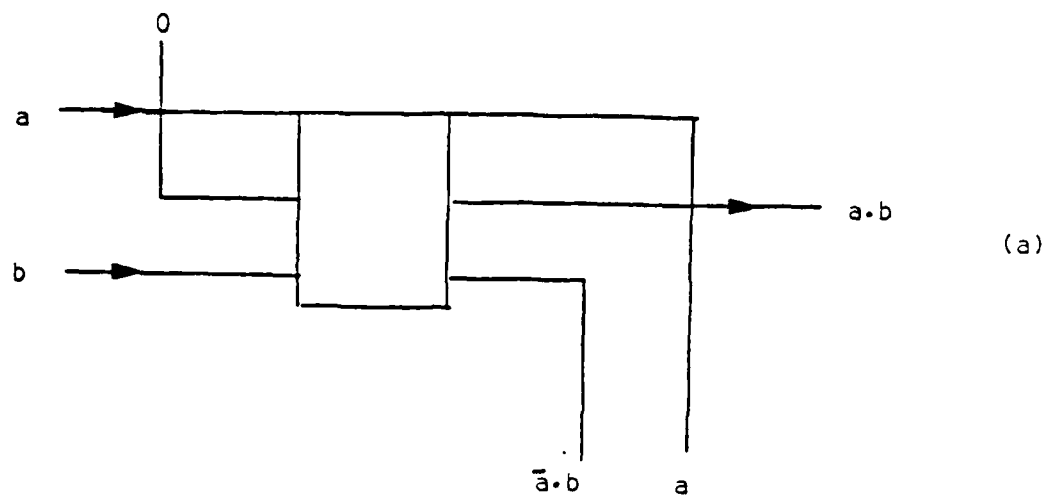


Fig. 2

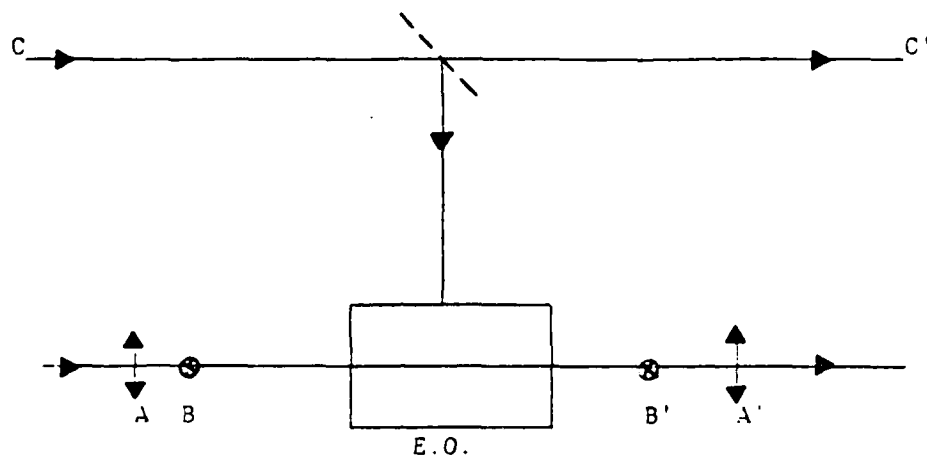


Figure 3.

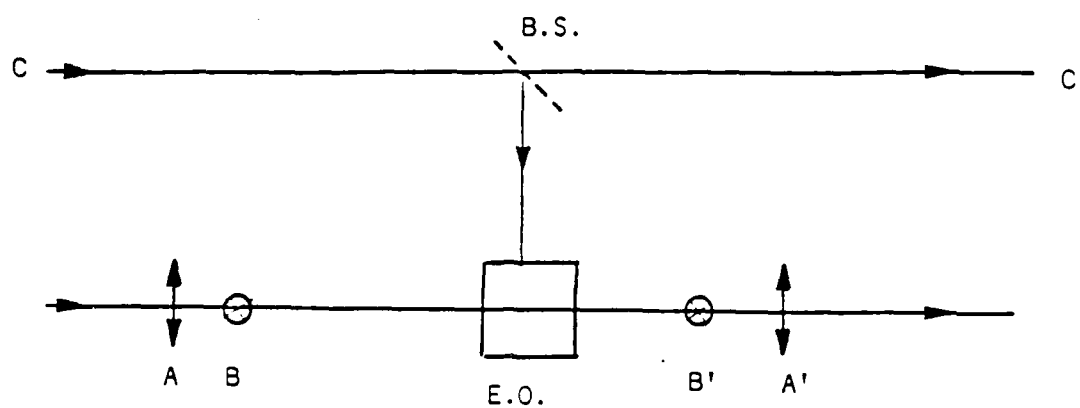


Fig. 3

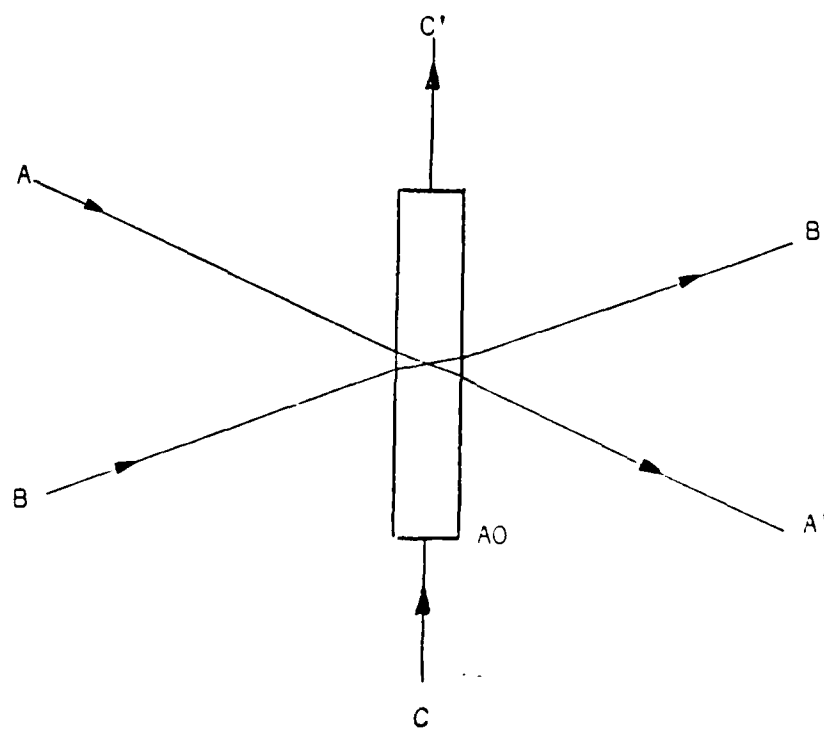


Fig. 4

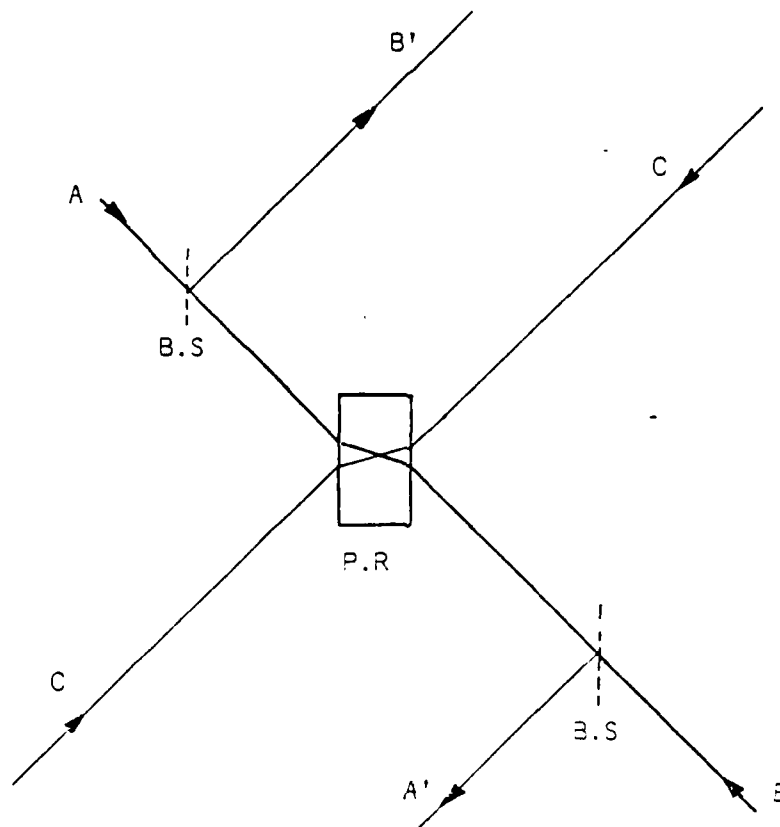
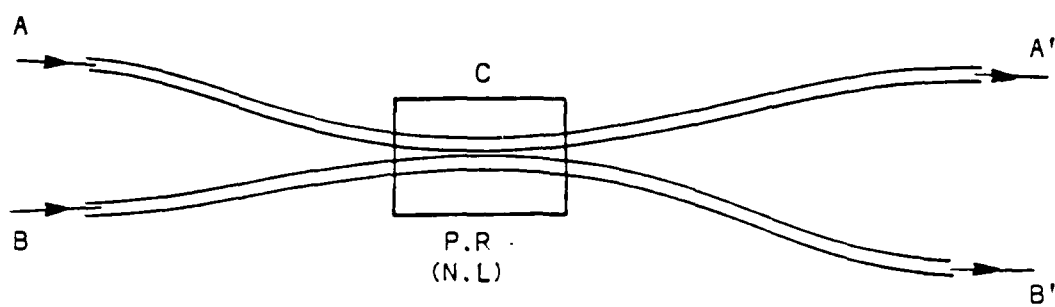
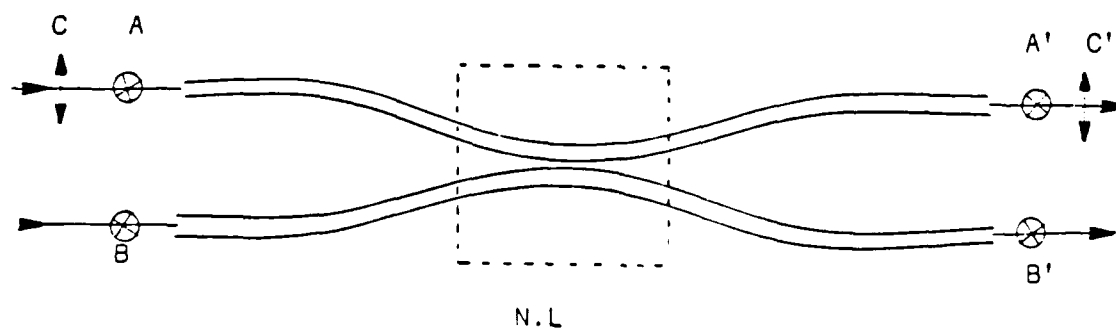


Fig. 5



(a)



(b)

Fig. 5



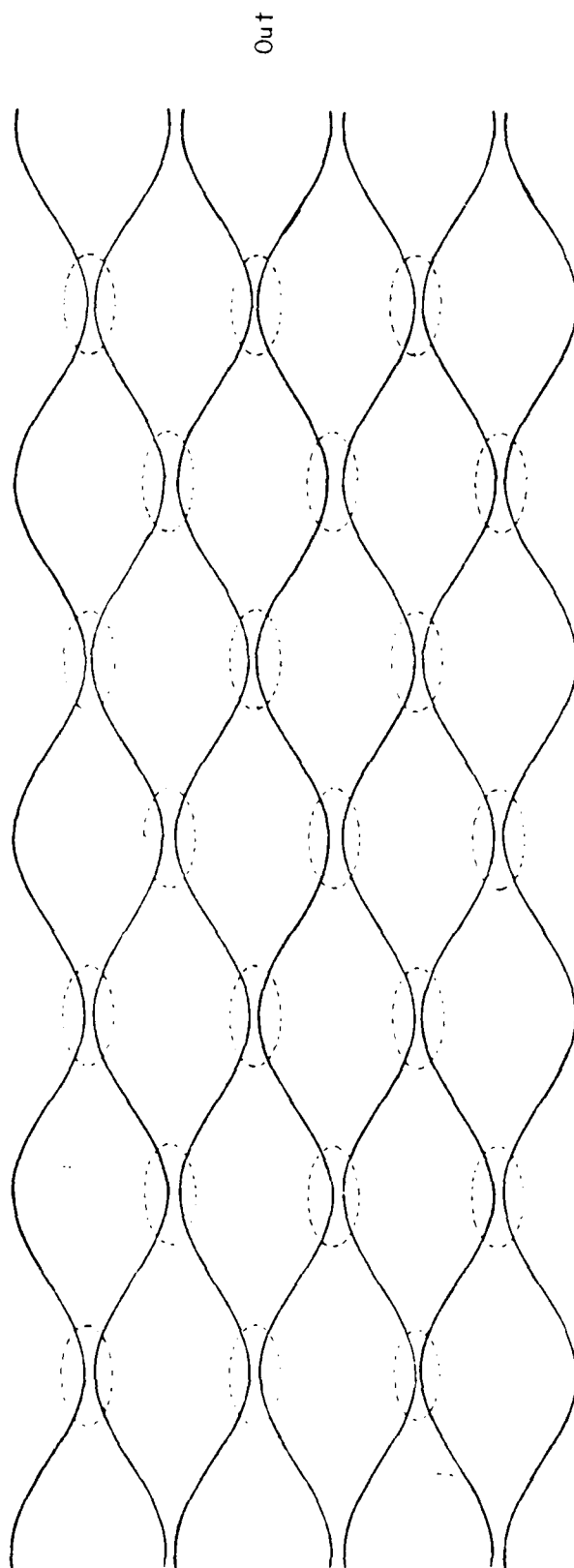
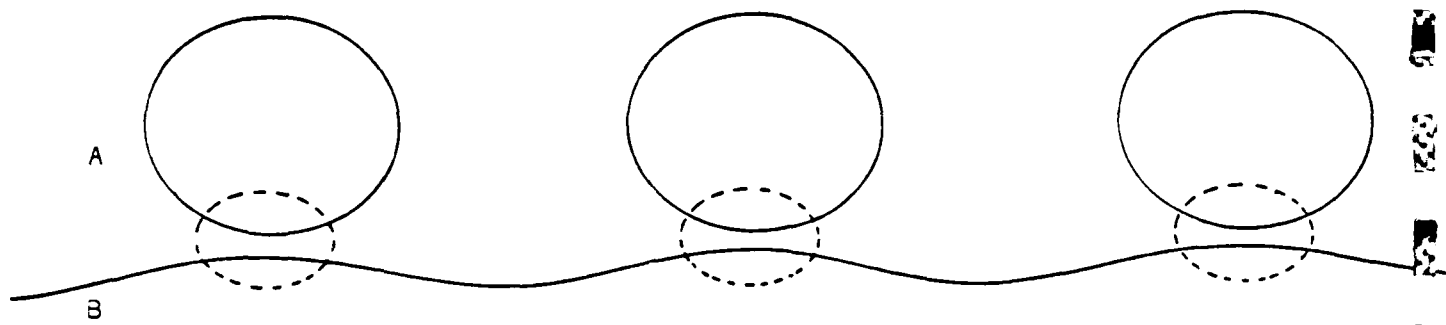
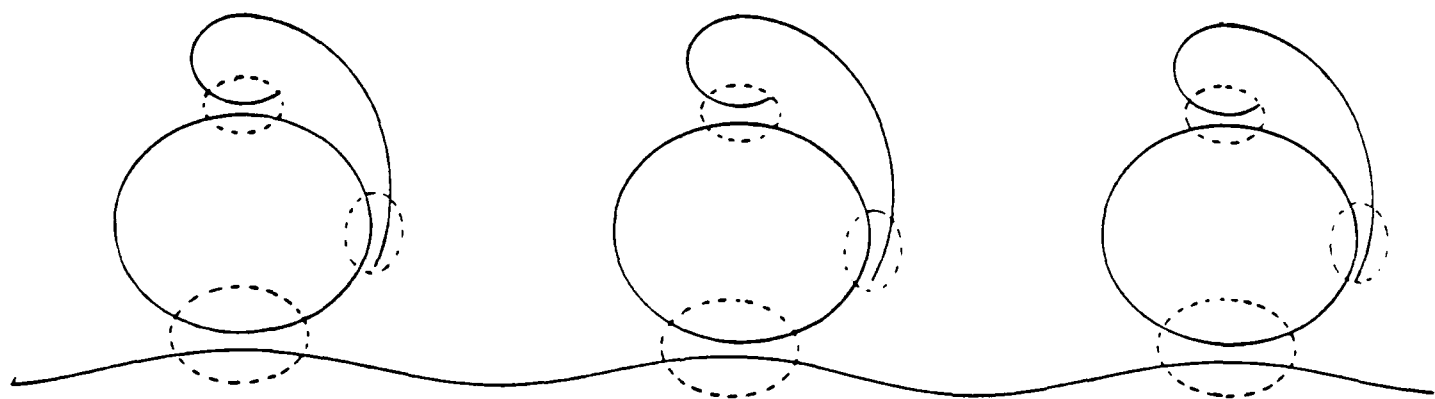


Fig. 7



(a)



(b)

Fig. 3

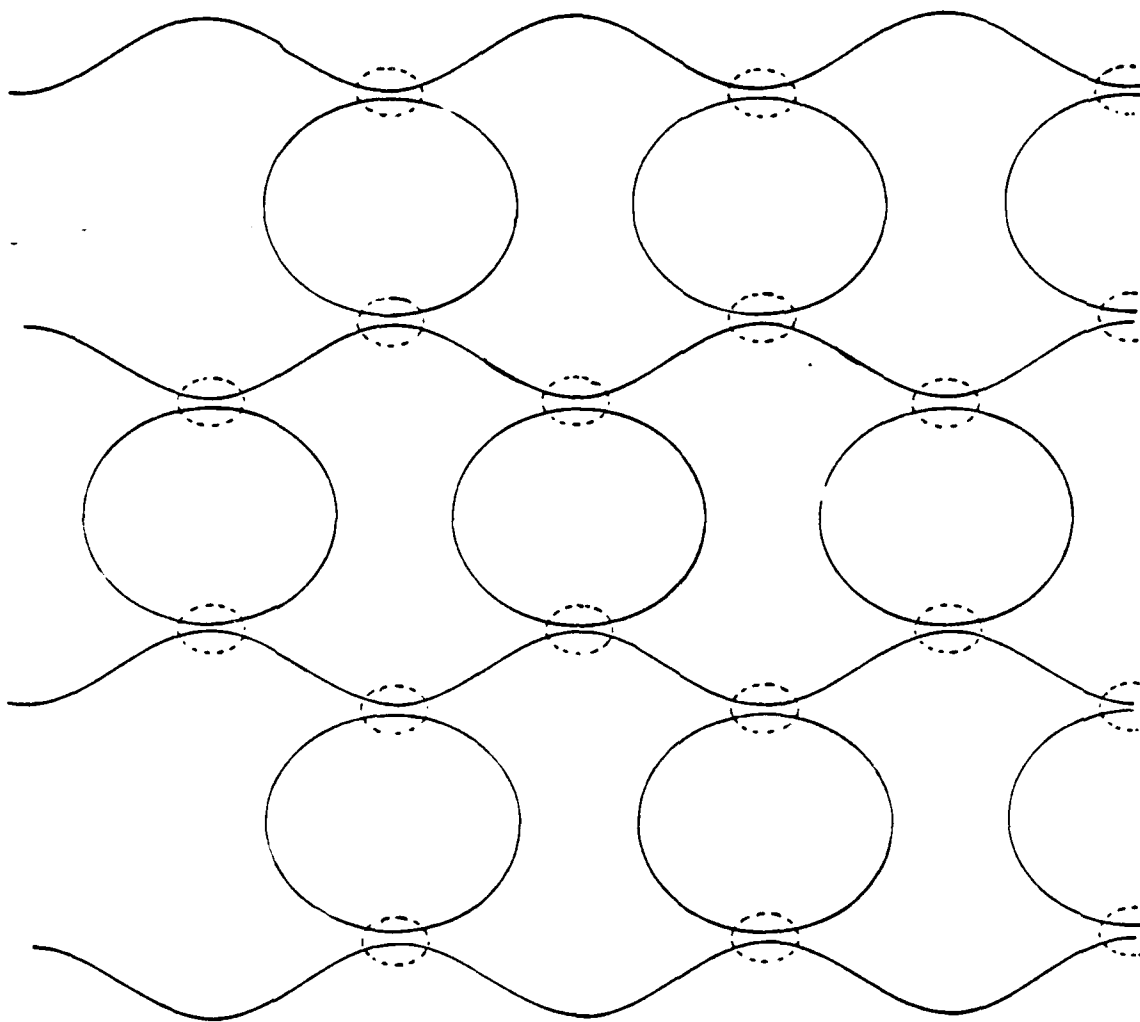


Fig. 8a

(b) Work in Progress

Work has been directed toward an optical neighborhood processor designed to operate on two dimensional array of 1's and 0's. The primary application will be to two dimensional image processing. We have achieved a design for an optically implemented table look-up device, which will be the heart of this processor. The architecture for this table look-up device is illustrated in Figure 1.

Input to this device consists of a 9-bit string of 1's and 0's. In image processing applications, this string will originate from a 3X3 neighborhood. Output from the device is a 1 or a 0, depending on whether the string is in the table or not. As shown in Fig. 1, the input is fed into two banks of 9 LED's. In one bank, an LED is lit in each position corresponding to a 1 in the input string. In the other bank, the positions corresponding to 0's only are lit. There are two tables of masks. In the first mask table, positions corresponding to 1's are clear, allowing light to pass through, while in the second, positions corresponding to 0's are clear.

Light from the LED's passes through the mask tables. A maximum amount of light passes through when the lit LED's coincide with the clear mask positions. Thus, all table entries which have 1's in the same positions as the input string will allow a maximum amount of light to pass through in the "1's channel" of the device. Similarly, table entries which have 0's in the same positions as the input string will allow a maximum amount of light through in the "0's channel". However, only the one table entry which has both 1's and 0's in the same positions as the input string (i.e., which exactly matches the input string) will allow the maximum amount of light (the intensity of 9 LED's) through both channels at the same time.

Light which passes through the mask tables is collected onto detectors, summed together, and fed into a threshold device. The threshold can be set, for example, at 8.5, so that only the light from 9 LED's will be greater than the threshold. Since this amount of light will pass through only when the input string exactly matches a table entry, the threshold device will determine when a match has occurred.

By transforming 3X3 array neighborhoods into 9-bit strings, this table look-up device can be used as cellular array processor for applications such as image processing. One possible transformation can be described as follows. Consider an array with coordinates labeled as shown:

(1,1)	(1,2)	(1,3)	(1,4)	...
(2,1)	(2,2)	(2,3)	(2,4)	...
(3,1)	(3,2)	(3,3)	(3,4)	...
.	.	.	.	.
.	.	.	.	.

The 3 3 neighborhood in the upper left corner can be transformed to the string

(1,3)  
 (2,3)  
 (3,3)  
 (1,2)  
 (2,2)  
 (3,2)  
 (1,1)  
 (2,1)  
 (3,1)

The next neighborhood to be transformed is obtained by moving one position to the right. This neighborhood is transformed to

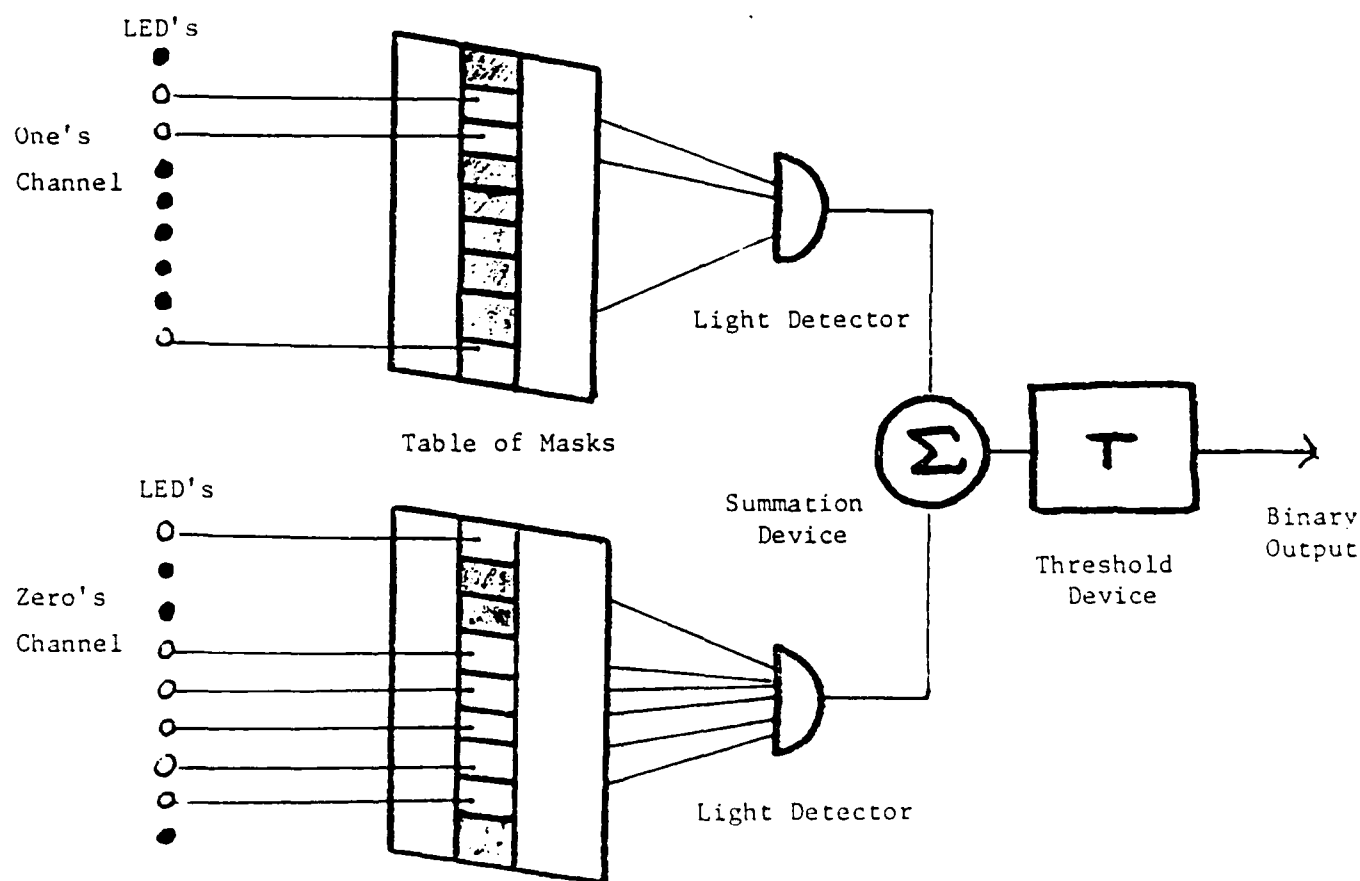
(1,4)  
 (2,4)  
 (3,4)  
 (1,3)  
 (2,3)  
 (3,3)  
 (1,2)  
 (2,2)  
 (3,2).

New information is added at the 'top' of the string, while old data drops out the 'bottom'. The advantage of this transformation scheme is that it allows for a smooth flow of data from the two dimensional array to the input bit string.

### 3. CONCLUSIONS

From these preliminary studies, some equally preliminary conclusions can be drawn. First, high accuracy results do not always require high accuracy processors. By clever use of appropriate electronics, we can combine some of the virtues of high speed analog optical processing with the accuracy normally associated with high speed digital processing. We conclude, as well that ultimately high speed optical processing may very well utilize optical Fredkin Gates. In any case, further study of this technique seems amply justified.

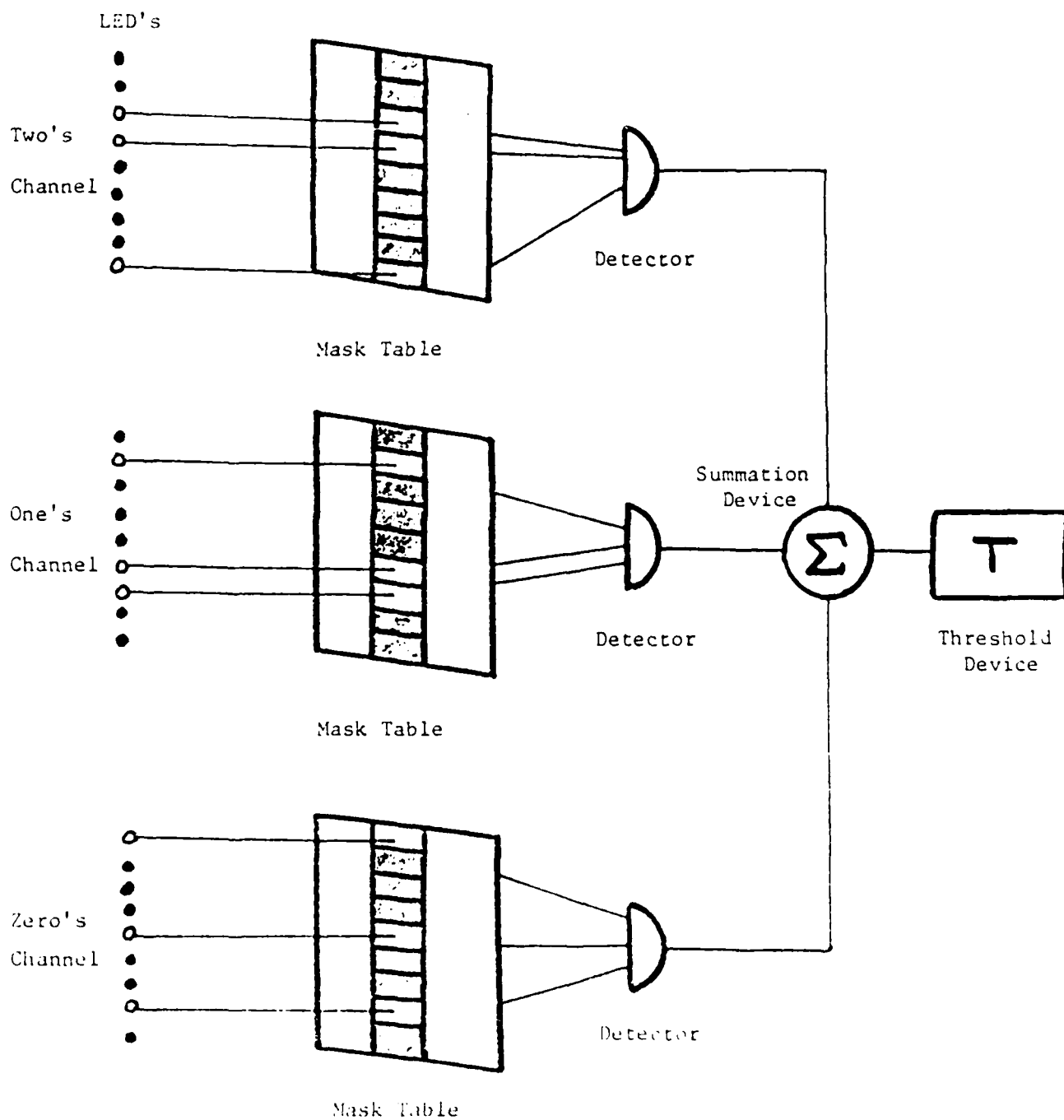
Example: Search the table for the binary string '011000001'. In the "one's channel", the LED's are lit with the pattern 'ULLUUUUUL' (U = Unlit, L = Lit), while in the "zero's channel" the pattern is 'LUUUUUUUU'.



ARCHITECTURE FOR TABLE LOOK-UP DEVICE (BINARY INPUT)

FIGURE 1

Example: Search the table for the ternary string '012201102'. In the "two's channel", the LED's are lit with the pattern 'UULLUUUUL', in the "one's channel" the pattern is 'ULUUULLUU', and in the "zero's channel" the pattern is 'LUUUUUUUU'.



ARCHITECTURE FOR TABLE LOOK-UP DEVICE (THREE LEVEL INPUT)

FIGURE 2



# SPEED AND ACCURACY OF THE BIMODAL OPTICAL COMPUTERS

Mustafa A.G. Abushagur  
Electrical and Computer Engineering Department  
and  
Center for Applied Optics  
University of Alabama in Huntsville

H. John Caulfield  
Center for Applied Optics  
University of Alabama in Huntsville  
Huntsville, AL 35899

December 1985

## ABSTRACT

An optical-hybrid computer is compared in its speed to the digital computers. Optical-hybrid computers are shown to be far more superior in their speed in solving systems of linear equations. This advantage in speed increases with the increase of the size of the matrix. The problem of the convergence of the solution using the optical-hybrid computer is discussed and it is found that using optical systems with an error of about 5% assures convergence for matrices with condition number as high as 150. Some means of improving the condition number of a matrix are also introduced.

## I- INTRODUCTION

Analog optics is very attractive for signal processing and computing because of its ability to process two-dimensional data in parallel very rapidly. Unfortunately, this high speed parallel processing achieves only low accuracy because of the nature of the analog processing especially in the optical systems. These accuracy problems rise from errors in representing and reading the signal using the electrooptic I/O devices. The method introduced by Caulfield<sup>(1)</sup> (which is described in the first part of this report) combines the high speed and parallelism of the optical computer and the high accuracy of the digital computer, using Lord Kelvin's iterative method<sup>(2)</sup>. In section II of this paper we present a comparison between the time required to solve a system of linear equations using the optical-hybrid computer versus that required by the digital computer. In section III we present a numerical analysis of the convergence of the solutions for a linear algebraic

equations as a function of the condition number of the matrix by using computer simulation of the optical-hybrid computer. In section IV a conclusion and final remarks are presented.

## II. COMPUTATION SPEED ANALYSIS

The optical-hybrid computer works in the following manner for a system of linear equations<sup>\*</sup>

$$A \underline{x} = \underline{b} . \quad (1)$$

a) Using an optical analog processor we can calculate an approximate solution  $\underline{x}^0$  of the linear system, the superscript <sup>0</sup>'s indicate inaccuracies in the optics and electronics

$$A \underline{x}^0 = \underline{b}^0 . \quad (2)$$

b) Remember the solution to a high accuracy. Use a dedicated digital processor to calculate the residue

$$\underline{r} = \underline{b} - A \underline{x}^0 = A (\underline{x} - \underline{x}^0) = A \Delta \underline{x} \quad (3)$$

c) Use the optical analog processor to solve the linear equation

$$A \underline{y} = s \underline{r}^0 , \text{ where } \underline{y} = s \Delta \underline{x} , \quad (4)$$

for  $\Delta \underline{x}$ , where  $s$  is a "radix", or scale factor chosen to make a good use of the dynamic range.

d) Use the digital processor to refine the solution for  $\underline{x}^1$

$$\underline{x}^1 = \underline{x}^0 + \Delta \underline{x} . \quad (5)$$

---

\* It is also applicable to other problems - both linear and nonlinear.

If the refined solution  $\underline{x}^1$  is accurate enough terminate the iterations. Otherwise go back again to (b), (c) and (d) for a more refined solution following the above outlined procedure.

To get some quantitative values for the speed of this process compared to that carried by digital computers, we will calculate the number of operations required by each method then multiply it by the time required by each operation.

let us consider an  $n \times n$  matrix  $A$ , the time required for one iteration outlined above,  $Q_{O1}$ , is given by<sup>(3)</sup>

$$Q_{O1} = 2Q_{A1} + (n^2 + 3n)Q_{D1}, \quad (6)$$

where  $Q_{A1}$  = the time required to  $A\underline{x} = \underline{b}$  by analog optics,

$Q_{D1}$  = the time required to make one digital operation.

Therefore the time required to make  $I$  iterations is given by

$$Q_O = IQ_{O1} = I \{ 2Q_{A1} + (n^2 + 3n)Q_{D1} \}. \quad (7)$$

While the time required by the digital computer to solve the linear equation using the Gaussian elemination method takes  $n^3/3$  operations, and by using the Cholesky's method the number of operations can be reduced to  $n^3/6$ . Hence the time required to solve the linear equation,  $Q_D$ , is given by

$$Q_D = \frac{n^3 Q_{D1}}{6} \quad (8)$$

Comparing Eqs. (7) and (8) it is clear that

$$Q_O \ll Q_D, \quad (9)$$

Therefore, for a clear time advantage for the hybrid scheme, we

want

$$I \{ 2Q_{A1} + (n^2 + 3n) Q_{D1} \} \ll (n^3/6) Q_{D1} \quad (10)$$

or

$$2I Q_{A1} \ll [n^3/6 - I(n^2 + 3n)] Q_{D1} \quad (11)$$

or

$$\frac{n^3/6 - I(n^2 + 3n)}{2I} \frac{Q_{D1}}{Q_{A1}} \gg 1 \quad (12)$$

The advantage of using the optical-hybrid over the digital computer in speed is very obvious from Eq.(12), and it will increase by the increase of the size of the matrix  $n$ . Eq.(12) can be rewritten in the following format

$$A_P A_I \gg 1 \quad (13)$$

$$\text{where } A_P = \{ n^3/6 - I(n^2 + 3n) \} / 2I \quad (14)$$

$$A_I = Q_{D1} / Q_{A1} \quad (15)$$

Here  $A_I$  is an "inherent advantage". A single analog operation is much faster than the digital one. The whole  $Ax=b$  solution will be slower than a single digital operation, but the analog optical  $Ax=b$  solver works at speeds independent of  $n$ . On the otherhand,  $Q_{D1}$  is eigenvalue dependent.  $A_P$  is a problem related advantage with the increase in  $n$ ,  $A_P$  increases very rapidly. Clearly, also we want to keep the number of iterations low.

From the above discussion we see that the optical-hybrid computer can achieve results in a much shorter time especially for large matrix sizes. But does this process always work or converge?

### III CONVERGENCE OF THE SOLUTION

The block diagram of the optical-hybrid computer is shown in Fig.1. The solution of the linear algebraic equation will be done optically using the method introduced by Cheng and Caulfield<sup>(4)</sup>. The question of the convergence is discussed in the previous paper and it is found that if the matrix has positive eigenvalues then the solution will converge regardless of the size of the matrix. This of course, applies simply to step (c). We turn next to the total process.

In this section of the paper we present a numerical analysis of the convergence of the solution and its dependence on the condition number of the matrix. The condition number of the matrix  $A$  is defined as

$$k(A) = \|A\| \|A^{-1}\| \quad (16)$$

where  $\| \cdot \|$  is the norm of the matrix. The condition number is a measure of the accuracy of the  $Ax=b$  solutions. The larger the condition number the less accurate the result achieved with any fixed accuracy computer. In this paper we report a simulation of the system shown in Fig.1 by a computer algorithm to study the convergence of the solution of the linear equation. The computer algorithm simulates the analog optical processor and the electro-optic I/O devices in such a way that allows us to control the errors occurring in representing the matrix by an optical mask, and also the error in reading the photodiode voltage and in converting the voltage input to the system to light in the LEDs. To simulate the

experimental environment we have used a Gaussian random number generator to generate the error signals.

The curve shown in Fig.2 is the result of a simulation experiment for optical-hybrid computer with the following characteristics: The matrix  $A$  can be represented by an optical mask (a photographic film, or a spatial light modulator) with an error equal to 1% of the maximum coefficient of the matrix. The vector  $\underline{x}$  can be read with an error of standard deviation=1% of the maximum element of the vector  $\underline{x}$  in the electronics also the standard deviation in representing the vector  $\underline{b}$  by the photodiode is 1%. From Fig.2 we see that solutions converge with an error of less than  $10^{-6}$  (or any other accuracy) even for condition number 500. For condition numbers less than 250 the number of iterations required are less than 20. In order to guarantee convergence with 1% accuracies, we must restrict matrices to condition numbers less than 50.

To study the effect of the error in representing the matrix by an optical mask on the number of iterations to get a solution within  $10^{-6}$  error, we have changed the standard deviation of the error in representing the mask over the range from 1% to 30% for a condition number 150 and we calculated the number of the iterations required for each case. Fig.3 shows the relation between the number of iterations as function of standard deviation of the error in representing the matrix. As the error increases the number of iterations increase in an almost linear way. Even for an error of 30% in representing the matrix, the solution still converge. This interesting result proves that even by using inaccurate optics, the

optical-hybrid computer can still solve the linear equation very accurately. This result is experimental, and therefore, guaranteed. It appears, however, that this approach works often even in these unguaranteed cases.

The condition number is one the determining factors of the speed of convergence of the solutions as can be seen from Fig.3. Smaller condition numbers yeild faster convergence of the solution. In searching for a way to improve the condition number of a given matrix, we found one way of doing that is by normalizing the matrix in the following manner

$$a_{i1} = a_{i1} / [a_{i1}^2 + a_{i2}^2 + \dots + a_{in}^2]^{1/2} \quad ; i=1,2,\dots,n \quad (17)$$

where a is the coefficient of the matrix A . This normalization decreases the value of the condition number of the matrix which in turn increases the speed of the convergence process. Fig.4 shows a plot of the condition number before and after the normalization of the matrix, from which we can see an improvement in the condition number after the normalization.

#### IV CONCLUSIONS

The optical-hybrid computer discussed in this paper have shown very promising results, it is clearly faster than the digital computer in solving this class of problems. The advantage of the speed of this optical-hybrid increases with the increase of the size of the matrix. The analysis carried above is not limited to the



solution of a system of linear equations but is applicable as well to other linear and nonlinear problems. The same calculations we carried for the comparison of the speed is quite similar to that of the power consumption. Another interesting result presented here is that the optics which is used in the system can a tolerance of 5 to 10% without sacrificing the accuracy of the solution, although it is shown that the less error in both optics and electronics the faster the solution will converge.

## REFERENCES

1. H. John Caulfield, J.H. Gruninger, J.E. Ludman, K. Steiglitz, H. Rabitz and J. Gelfand, Bimodal Optical Computers, to be published.
2. W. Thompson (Lord Kelvin), Proc. Roy. Soc. (London) 28, 111 (1878).
3. G.W. Stewart, Introduction to Matrix Computations, Academic Press 1973.
4. Wai K. Cheng and H. John Caulfield, Fully-Parallel Relaxation Algebraic Operations for Optical Computers, Opt. Comm. 43, No. 4, 251 (82).

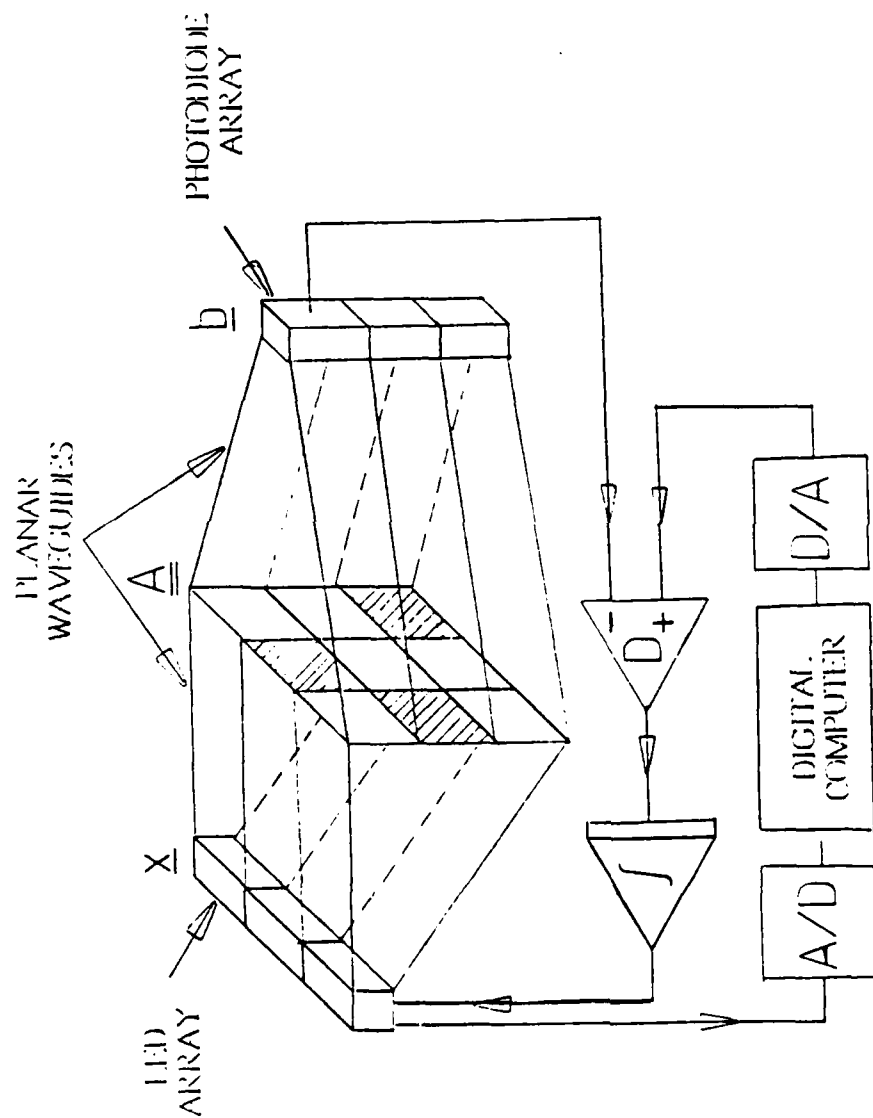


Fig.1 System layout of the optical-hybrid computer.

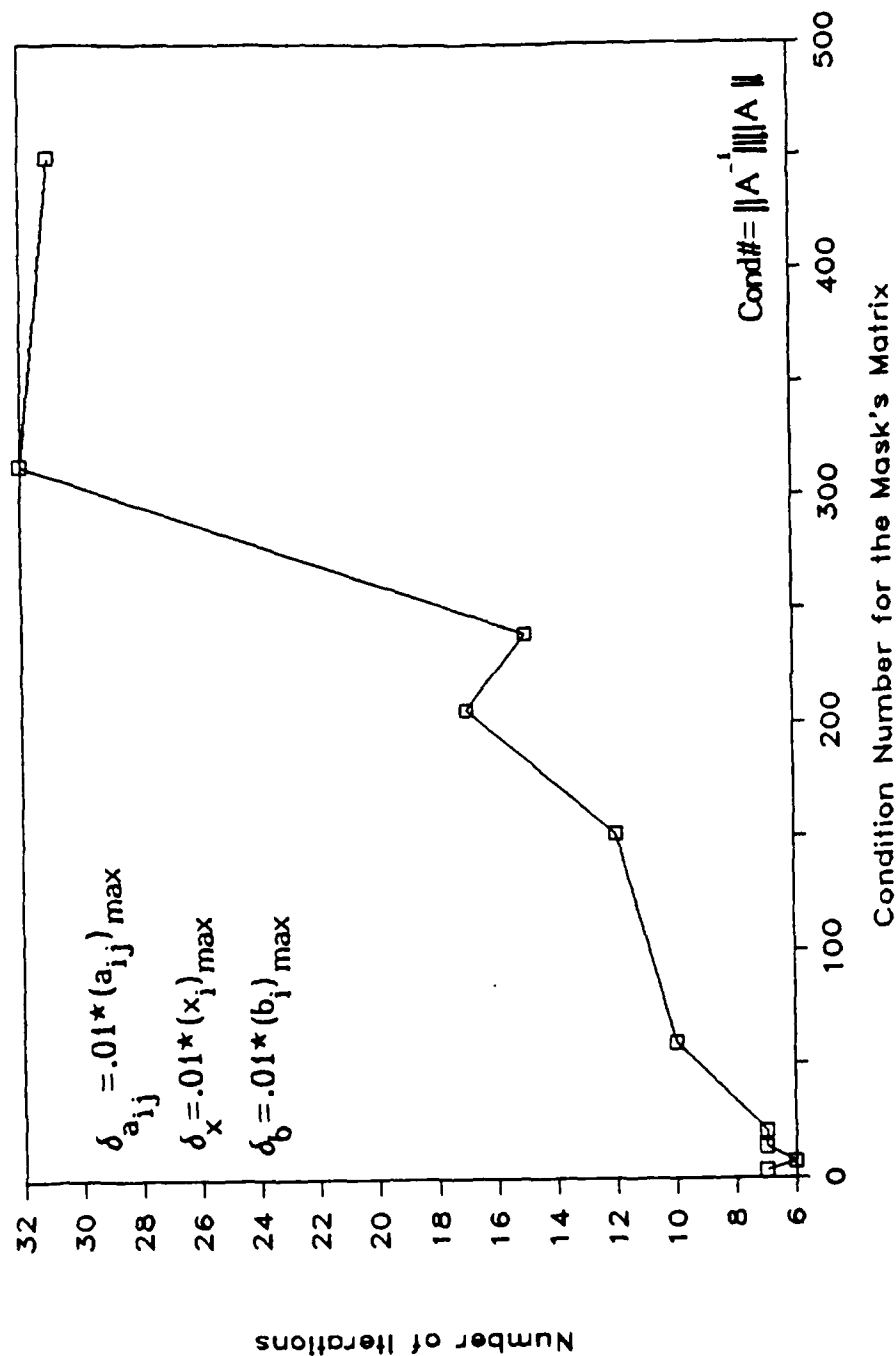


Fig.2 Computer simulation results for the number of iterations needed for the convergence of the solution are plotted as a function of the condition number of the matrix.

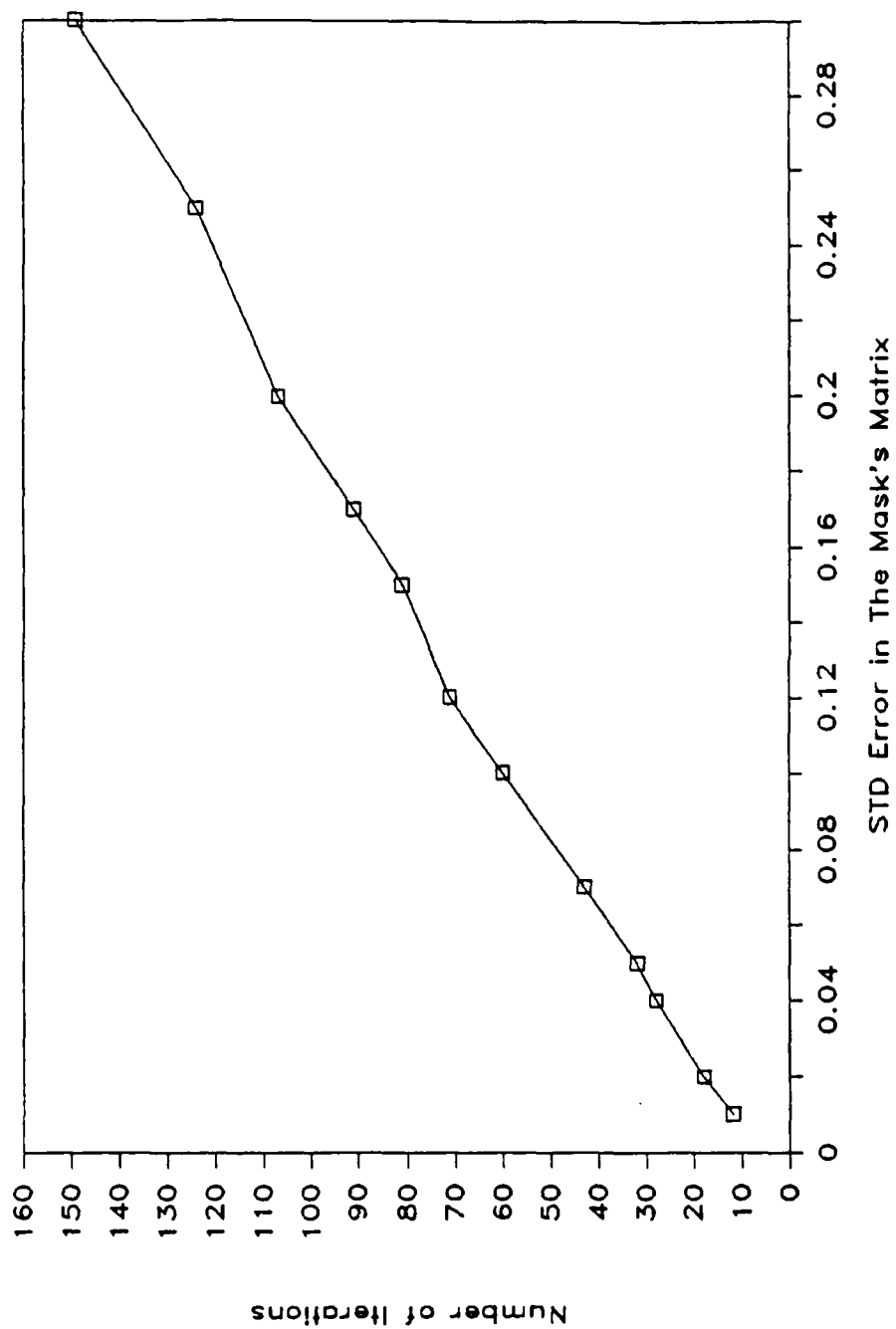


Fig.3 The number of iterations needed for the convergence of the resolution are plotted as a function of the standard deviation of the error in representing the matrix by the optical mask, for a matrix with condition number equal to 150.

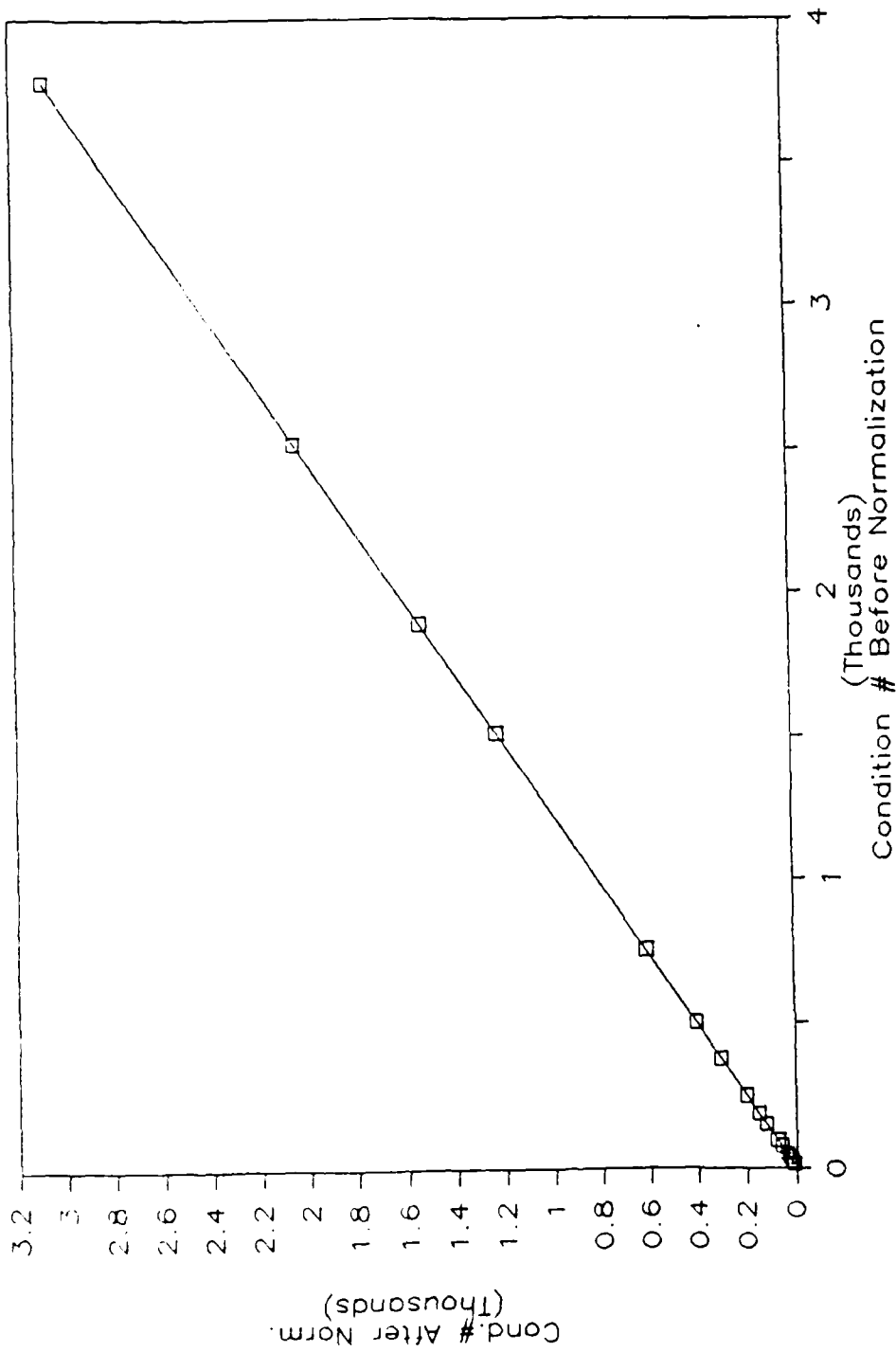


Fig.4 The condition number of the matrix after it has been normalized is plotted versus the original condition number.

**A-O COMPUTING STUDY**

**Jerry L. Erickson, Principal Investigator**

**PROBE SYSTEMS DIVISION  
ULTRASYSTEMS DEFENSE AND SPACE SYSTEMS, INC.**

PROBE SYSTEMS DIVISION  
ULTRASYSTEMS DEFENSE AND SPACE SYSTEMS, INC.  
655 North Pastoria Avenue  
Sunnyvale, CA 94086

A-O COMPUTING STUDY

FINAL REPORT

PS-ER-5652-01

31 December 1985



## TABLE OF CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
1.	OPTICAL J-K FLIP-FLOP ANALYSIS INTRODUCTION . . . . .	1
1.1	Optical J-K Flip-Flop Description . . . . .	1
1.2	J-K Flip-Flop Modes of Operation. . . . .	4
2.	SYNCHRONIZATION EFFECTS IN THE OPTICAL J-K FLIP-FLOP. . . . .	10
2.1	The Effects of Optical Path Length Errors . . . . .	10
2.2	The Effects of Asynchronization of Input Signals. . . . .	10
2.3	The Effects of Pulse Dispersion in the Thresholding Gain Mechanism . . . . .	10
2.4	Effects of Mismatch Between Input Period and Round Trip Delay	20
3.	THRESHOLD AND WEIGHTING ANALYSIS. . . . .	25
3.1	Noncoherent Optical J-K Flip-Flop Analysis. . . . .	25
3.2	Coherent Optical J-K Flip-Flop Analysis . . . . .	33

## LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1-1	Proposed Optically Implemented J-K Flip-Flop . . . . .	2
1-2	Relationship Between Inputs and Outputs of Ideal Thresholding Gain Device . . . . .	3
1-3	Ideal Operation of Optical J-K Flip-Flop . . . . .	6
1-4	Multiplexed Operation of Optical J-K Flip-Flop . . . . .	8
1-5	Optical J-K Flip-Flop Operating at 100% Duty Cycle . . . . .	9
2-1	The Effects of a +0.1% Error in the Length of The X Feedback Path . . . . .	11
2-2	The Effects of a -0.1% Error in the Length of The X Feedback Path . . . . .	12
2-3	The Effects of a +0.1% Error in the Length of The Y Feedback Path . . . . .	13
2-4	The Effects of a -0.1% Error in the Length of The Y Feedback Path . . . . .	14
2-5	The Effects of a +0.1% Error in the Length of The Z Feedback Path . . . . .	15
2-6	The Effects of a -0.1% Error in the Length of The Z Feedback Path . . . . .	16
2-7	The Long-Term Effects of a +0.1% Error in the Length of The Feedback Path. . . . .	17
2-8	The Effects of the J Input Leading The K Input By 1% . . . . .	18
2-9	The Effects of the J Input Lagging The K Input By 1%. . . . .	19
2-10	The Effects of Pulse Width Dispersion of 0.2%. . . . .	21
2-11	The Effects of Match Filtering on Spurious Signals Introduced by a 1% Error in The X Path Length. . . . .	22
2-12	The Effects of the Input Clock Rate 5% Higher Than The Internal Clock Rate. . . . .	23
3-1	Noncoherent Implementation of Optical J-K Flip-Flop. . . . .	26
3-2	Equivalent Model of Noncoherent Optical Interconnect and Threshold. . . . .	27

## LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
3-3	Probability Density Functions at Input to Threshold and Probability of Error Calculation For "OR" Operation . . . . .	30
3-4	Probability Density Functions at Input to Threshold and Probability of Error Calculation For "AND" Operation. . . . .	31
3-5	Equivalent Model of Coherent Optical Interconnect and Threshold . . . . .	36
3-6	Phasor Representation of Complex Weighting of Input Signal. .	38
3-7	Phasor Representation of the Sum of Two Arbitrary Signals With Noise Components. . . . .	40
3-8	Chi-Squared Density Function. . . . .	43
3-9	Optimum Weights and Threshold For X Logic Operation . . . . .	44
3-10	Optimum Weights and Threshold For Z Logic Operation . . . . .	45
3-11	Optimum Weights and Threshold For "OR" Logic Operation. . . .	46
3-12	Optimum Weights and Threshold For "Exclusive OR" Operation. .	47
3-13	Optimum Weights and Threshold For "AND" Logic Operation . . .	48

A-O COMPUTING STUDY  
FINAL REPORT

1. OPTICAL J-K FLIP-FLOP ANALYSIS INTRODUCTION.

1.1 Optical J-K Flip-Flop Description.

The proposed implementation of an optical J-K flip-flop is shown in Figure 1-1. In this implementation, the J and K inputs are externally supplied binary laser signals. The binary signal is represented by a on-off modulation of a laser carrier. These signals enter the flip-flop through a beam splitter and pass through a holographic interconnection array. The interconnection array provides a weighted interconnect between inputs and outputs as shown.

The output signals then pass through a thresholding gain mechanism. This mechanism amplifies signals of amplitudes greater than a built-in threshold. And signals below the threshold are not passed. Ideally the outputs of the thresholding gain mechanism are related to the inputs by the relations given in Equations (1-1) and (1-2) and shown in Figure 1-2.

$$\text{Input} > \text{threshold} \qquad \text{output} = 1 \qquad (1-1)$$

$$\text{Input} < \text{threshold} \qquad \text{output} = 0 \qquad (1-2)$$

In the case shown in Figure 1-1 a threshold of  $1/2$  is used. Some of the signals are fed back from the thresholding gain array to the holographic interconnect by a series of mirrors. This path not only provides feedback but also provides an inherent time delay that serves as an internal clock of the flip-flop. Finally, at the output of the thresholding gain mechanism, part of the output of the flip-flop exits through a beam splitter.

PS-86003

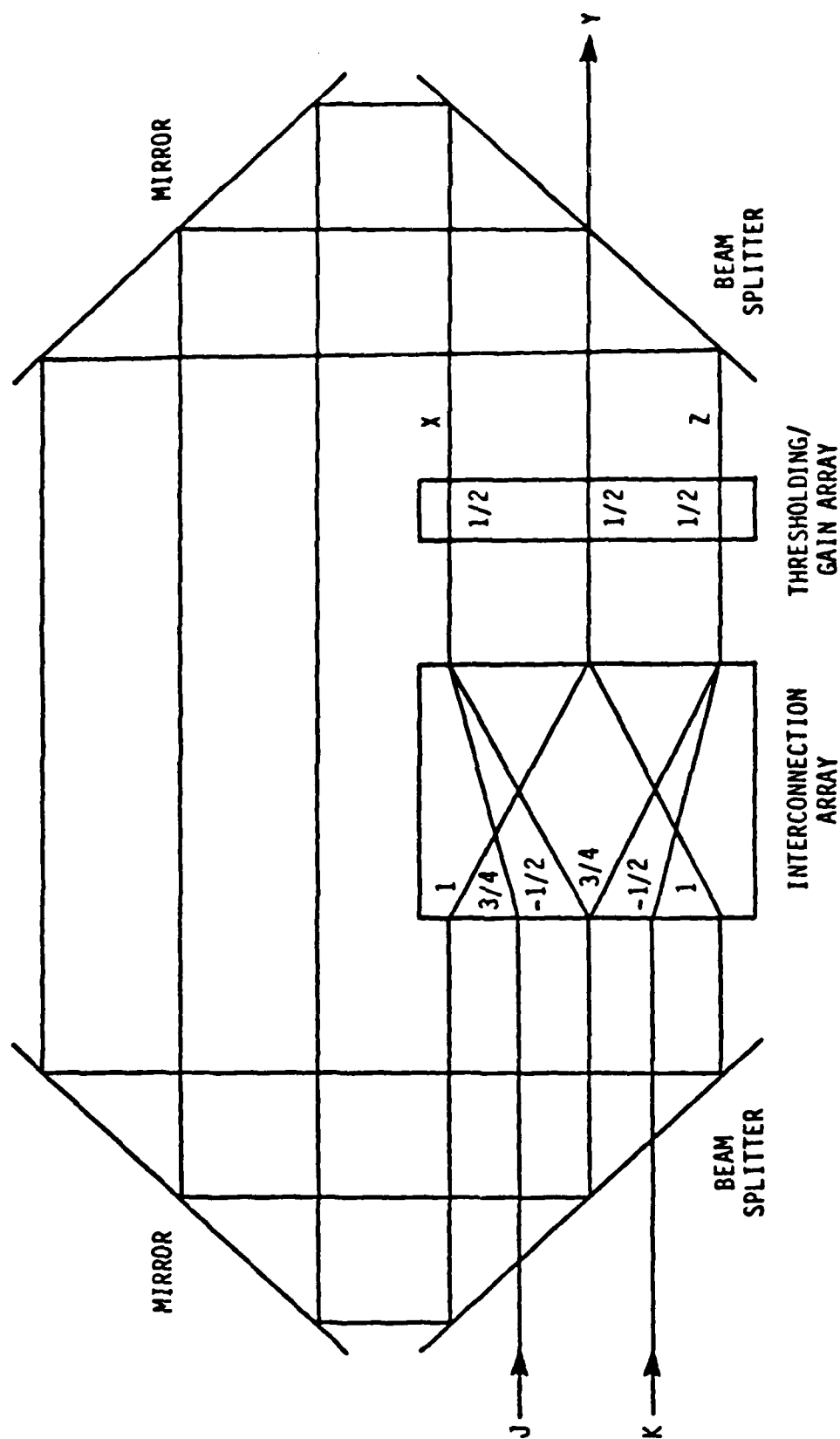
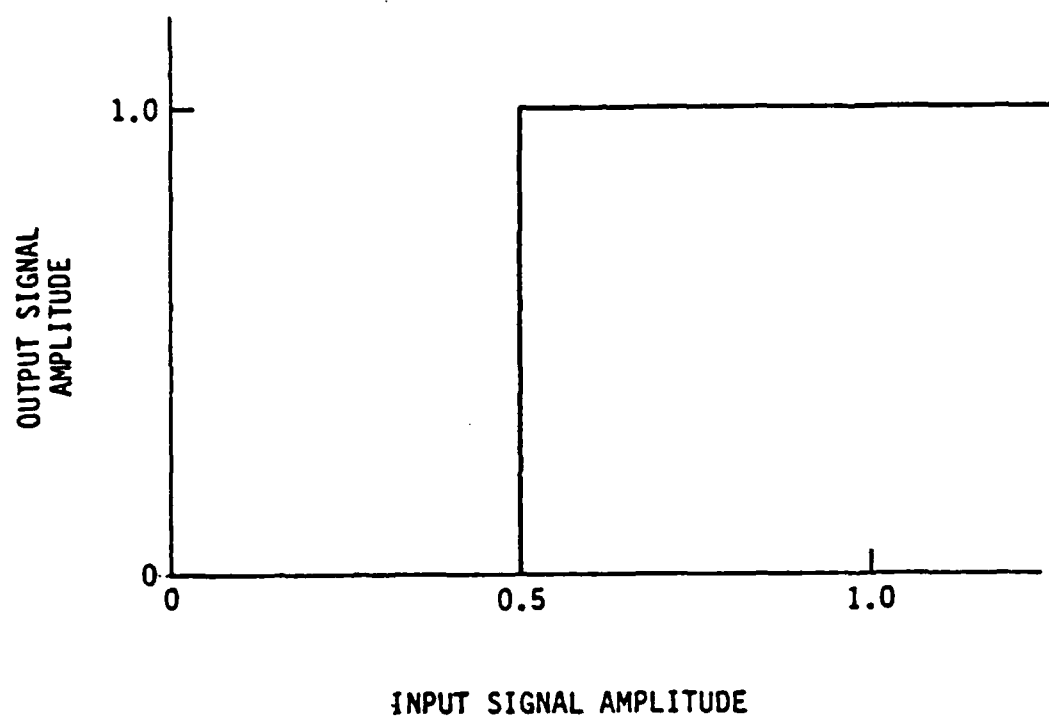


Figure 1-1. Proposed Optically Implemented J-K Flip-Flop



PS-86004

Figure 1-2. Relationship Between Inputs and Outputs of Ideal Thresholding Gain Device

### 1.1 -- Continued.

The key difference between the operation of this optical flip-flop and an electronic flip-flop is that the optical version lacks an externally supplied clock signal. The clock in the electronic flip-flop provides control over the sequencing of the state changes of the output of the flop-flop. Usually, flip-flops are designed such that the current state of the output is dependent on the state of the output at the last leading (or trailing) edge of the clock pulse. The truth table of an electronic J-K flip-flop is shown in Table 1-1.

Since the optical flip-flop lacks an external clock, the input signal period must be matched quite accurately to the internal feedback delay time. Since the feedback signals travel at the speed of light over relatively short paths ( $>1\text{m}$ ), the I/O rate of the optical flip-flop can be made many times faster than an electronic flip-flop.

### 1.2 J-K Flip-Flop Modes of Operation.

To begin the analysis, the J-K flip-flop was modeled under ideal conditions (no losses, no dispersion, use of the gain curve of Figure 1-2, equal optical feedback paths, and perfect synchronization of inputs with internal clocking) on a VAX 11/750, to verify the operation of the flip-flop. In addition, three modes of operation were simulated.

In the "basic" mode of operation, the J and K inputs are pulsed (e.g. 25% duty cycle) synchronously with the built-in delay. Figure 1-3 shows a timing diagram of the simulated inputs and outputs of the optical J-K flip-flop. First, the figure shows the case where both the J and K inputs are clocking a steady stream of "ones", and the Y output cycles alternately between "one" and "zero" (note: the outputs are delayed by  $1/2$  cycle). The remaining cases shown in Figure 1-3 are: (1) J-high and K-low (Y high), (2) J-low and K-high (Y low), and (3) J and K low (latches last value of Y into flip-flop). In comparison with the truth Table 1-1, the optical J-K flip-flop performs as predicted.

Table 1-1. J-K Flip-Flop Truth Table

J	K	$Y_i$	$Y_{i+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



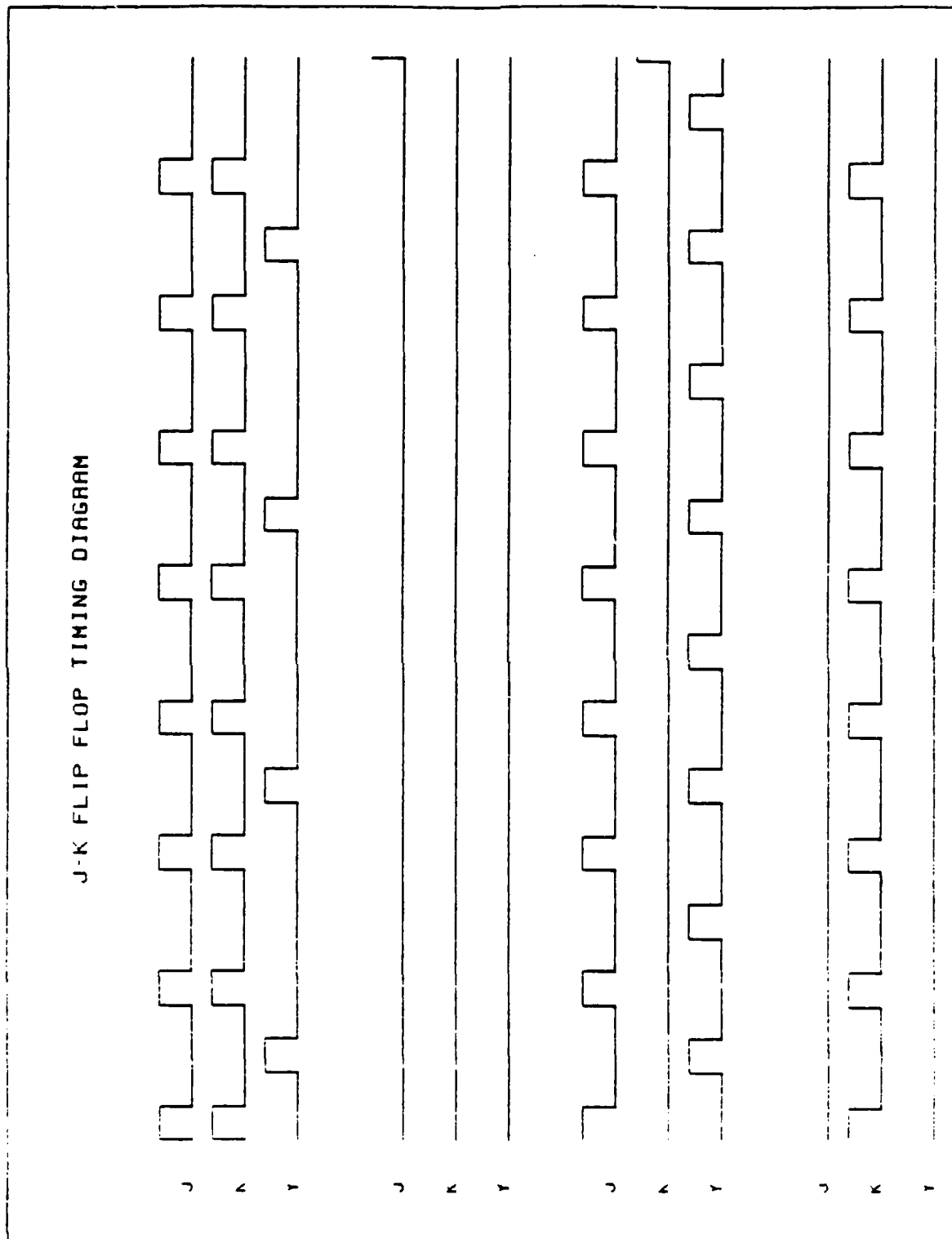


Figure 1-3. Ideal Operation of Optical J-K Flip-Flop

1.2      -- Continued.

Since the state of the output is only dependent on the state of the inputs one-half clock cycle back and is entirely independent of the state of the inputs anywhere else, the flip-flop can be operated in a time multiplexed mode, where several bits are input within a single feedback delay. Figure 1-4 shows an example of this mode of operation. In this figure the four inputs are time multiplexed into the flip-flop. The pulse width of any one of the inputs is  $1/8$ th the internal delay period. In the figure, the first and third inputs are held high and thus the first and third outputs cycle between high and low. The second inputs, show the J input high and the K input low, thus the output high. The fourth input, shows the J low and the K high, thus the output low. Again the multiplexed operation can be verified by the J-K flip-flop truth table (Table 1-1).

Finally, since the flip-flop is not constrained to operate on the leading or trailing edge of a clock pulse, a mode of operation can be hypothesized where the inputs are pulsed at a 100% duty cycle. Figure 1-5 demonstrates this mode of operation with inputs analogous to those used in Figure 1-3. Again, the operation of the J-K flip-flop can be verified using the truth table (Table 1-1).

In each of the cases discussed above, the operation speed is set by the round trip delay time (trt) of the feedback paths. The time between subsequent data entries must be equal to  $2 \times \text{trt}/N$  for N level multiplexing ( $N=1$  for nonmultiplexed operation).

## J-K FLIP FLOP TIMING DIAGRAM

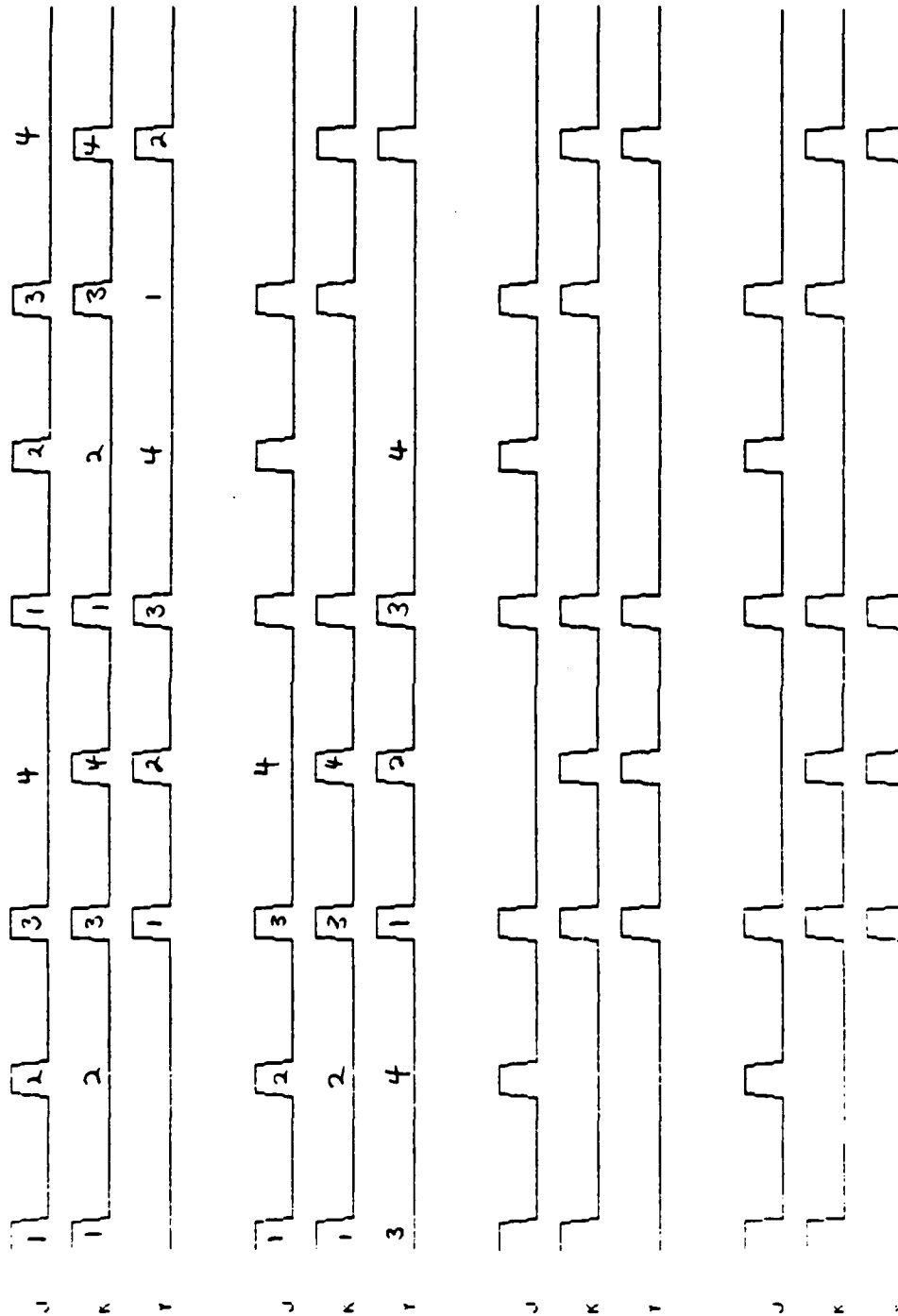


Figure 1-4. Multiplexed Operation of Optical J-K Flip-Flop

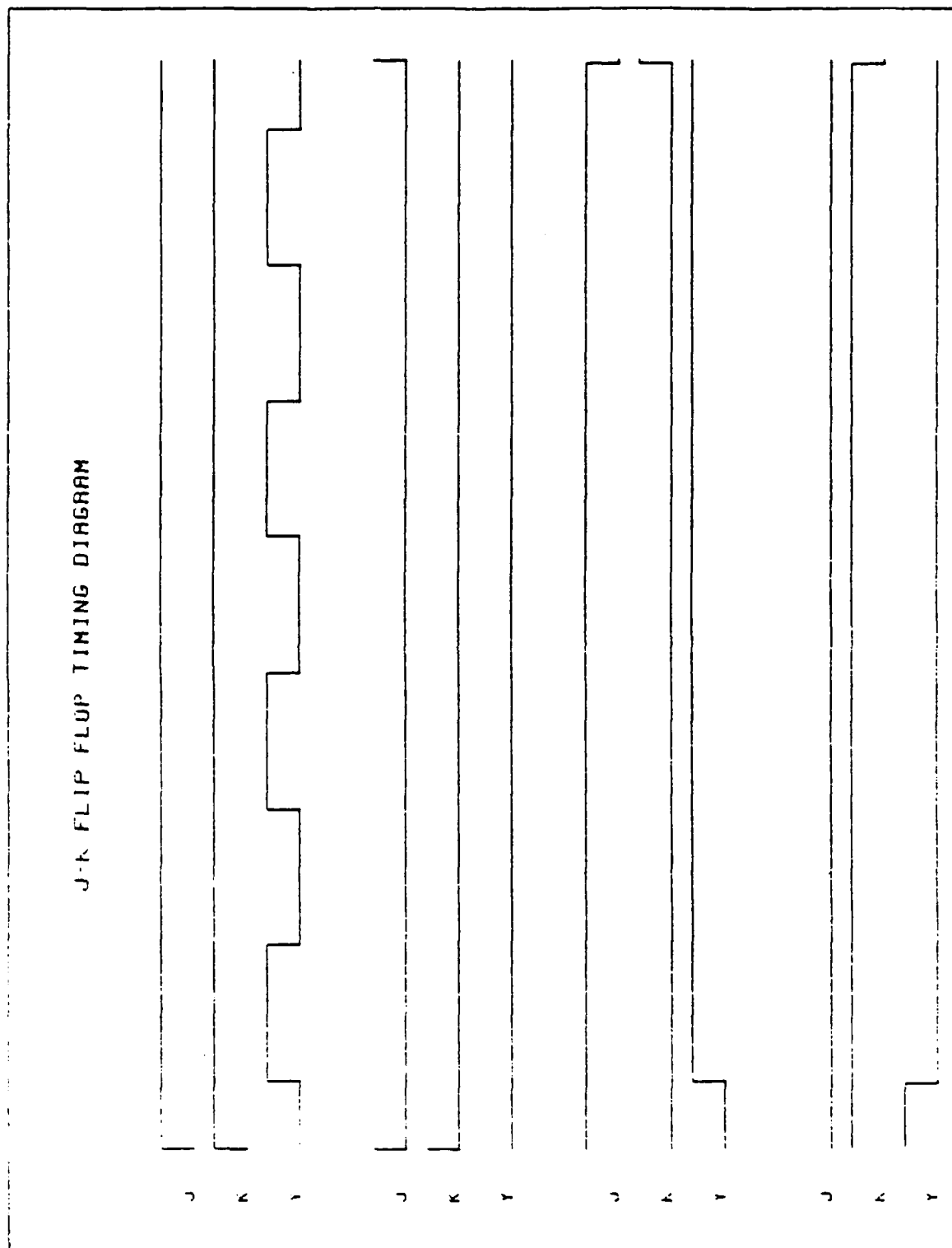


Figure 1-5. Optical J-K Flip-Flop Operating at 100% Duty Cycle

## 2. SYNCHRONIZATION EFFECTS IN THE OPTICAL J-K FLIP-FLOP.

This section discusses the effects of the asynchronous arrival of signals at the threshold gain mechanism. This section does not address the effects of phase errors on the destructive coherent interference used to obtain negative weights in the coherent system. The discussion of phase errors is deferred to Section 3.2. This section only addresses the more general and "idealized" effects of synchronization, and probably pertains more to the noncoherent case discussed in Section 3.1.

### 2.1 The Effects of Optical Path Length Errors.

The proposed optical J-K flip-flop contains three optical feedback paths (labeled X, Y, and Z in Figure 1-1). Errors in the lengths of any of these paths would introduce asynchronization of the arrival of pulses at the thresholding gain mechanism. The effect of the asynchronous arrival of pulses at the gain mechanism, is the introduction of spurious "glitchy" signals at the output of the flip-flop. Examples of the spurious signals introduced by errors of  $\pm 0.1\%$  in each of the optical feedback paths, (X, Y, and Z) are shown in Figures 2-1 through 2-6. In general, the narrow "glitches" can be minimized by internal or external filtering. However, as shown in Figure 2-7, in cases where the inputs remain fixed over many cycles, the glitches spread in time and become more apt to produce errors at the output.

### 2.2 The Effects of Asynchronization of Input Signals.

Another possible source of asynchronization of pulses at the gain mechanism is any skew of the input signals. These effects are shown in Figures 2-8 (J leads K by 1%) and 2-9 (J lags K by 1%). In both these cases we again see the introduction of spurious outputs.

### 2.3 The Effects of Pulse Dispersion in the Thresholding Gain Mechanism.

Another potential source of undesirable spurious signals is dispersion due to the finite frequency response of the thresholding gain mechanism. The finite frequency response of the thresholding gain mechanism will, in general,

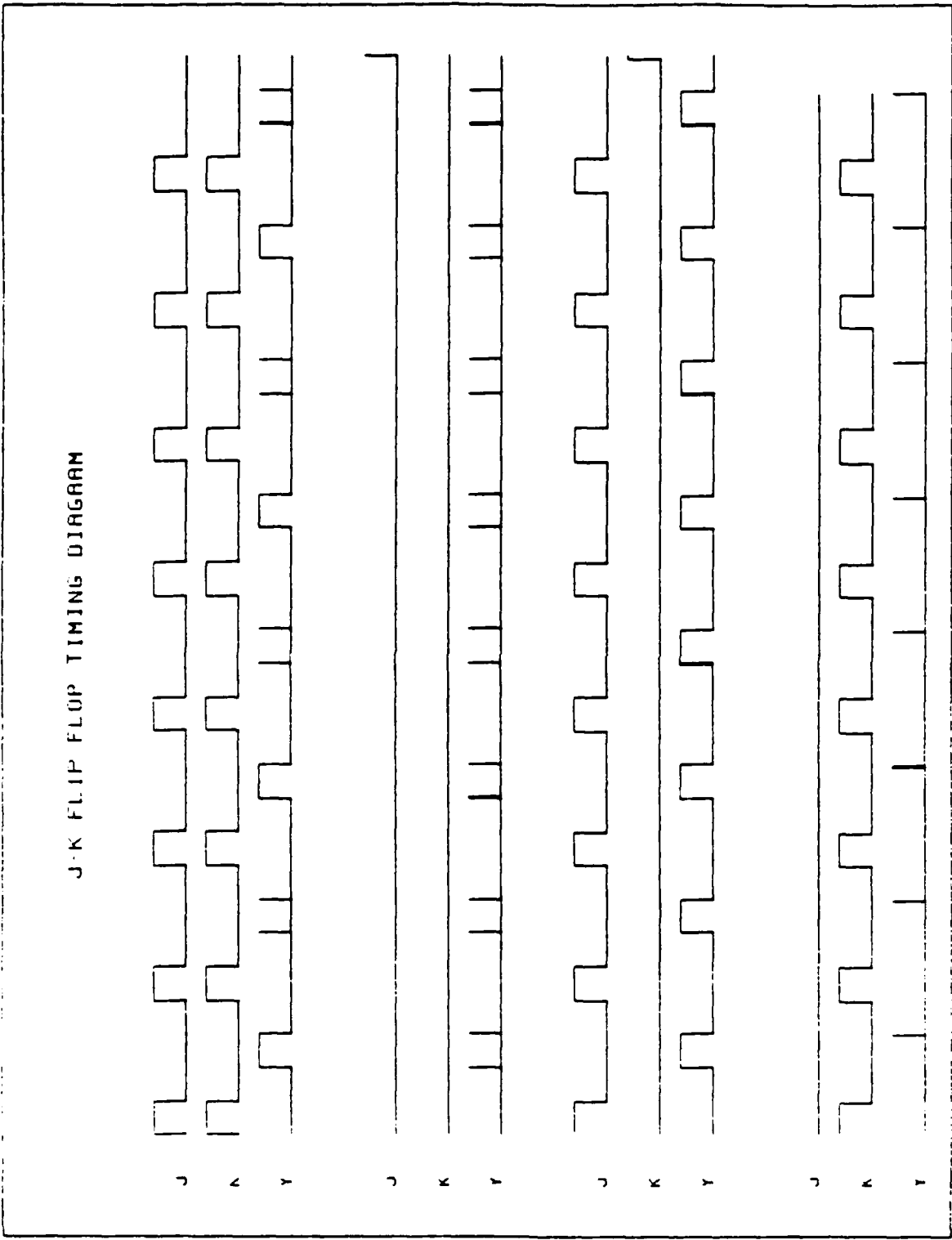


Figure 2-1. The Effects of a +0.1% Error in the Length of The X Feedback Path

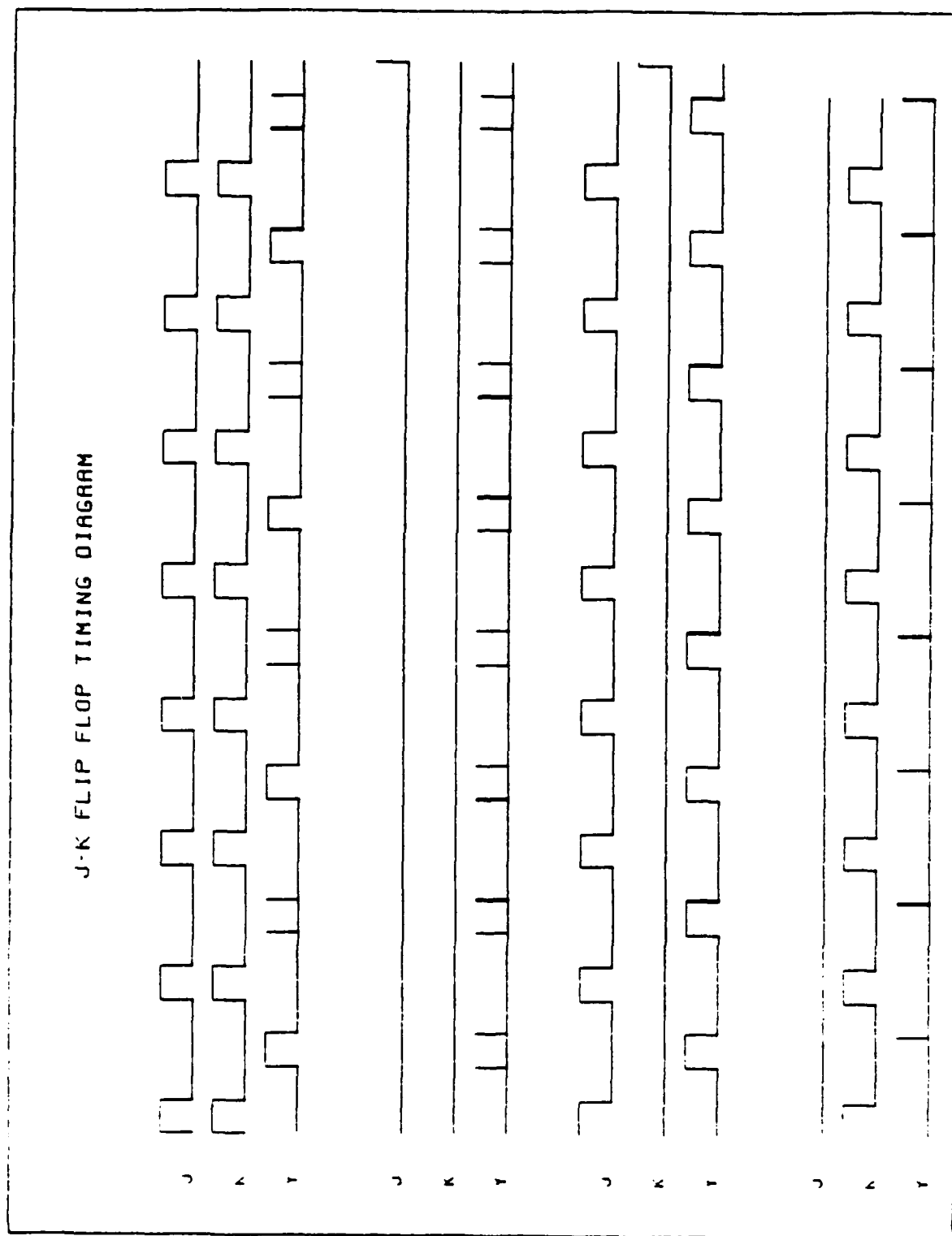


Figure 2-2. The Effects of a -0.1% Error in the Length of The X Feedback Path

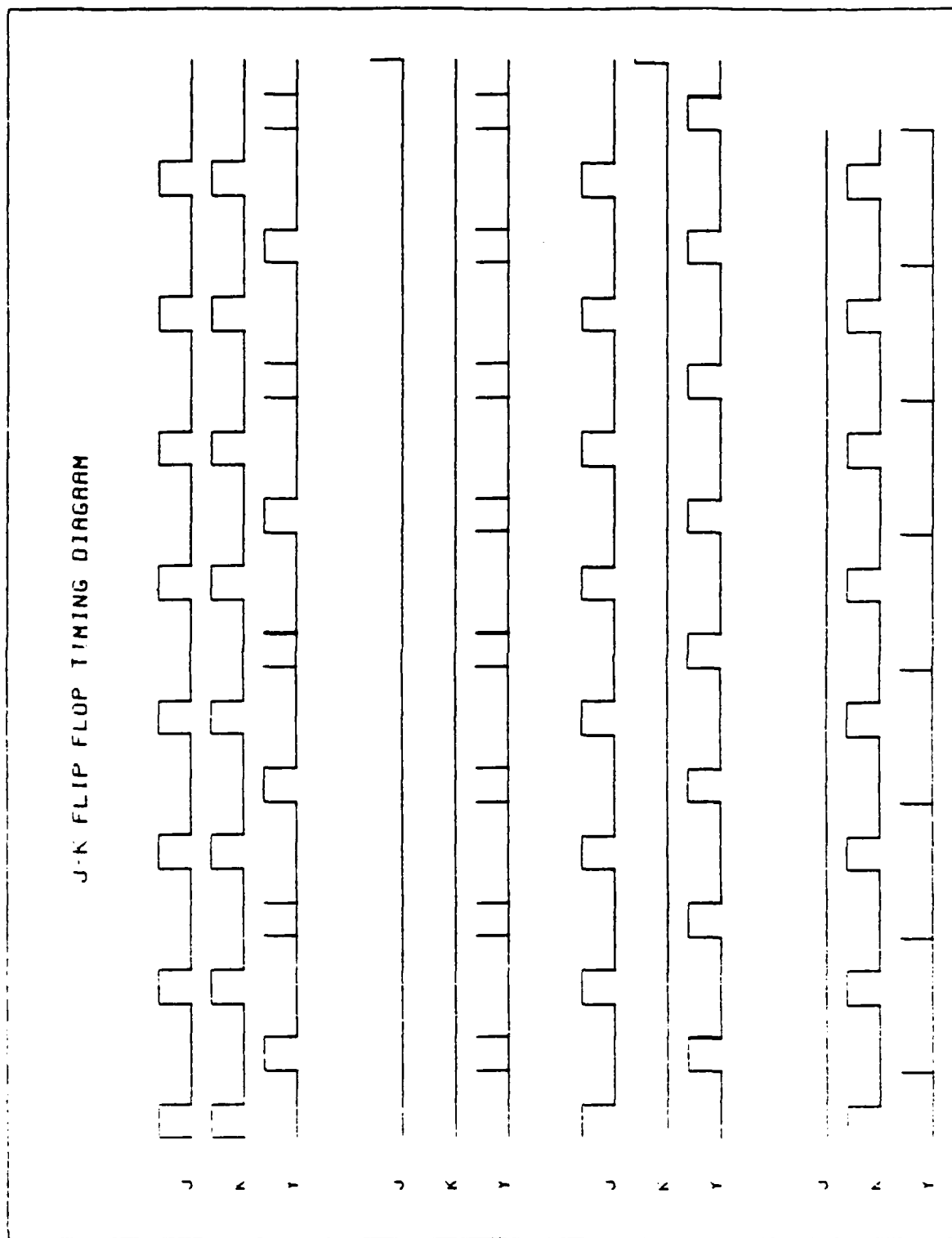


Figure 2-3. The Effects of a +0.1% Error in the Length of The Y Feedback Path



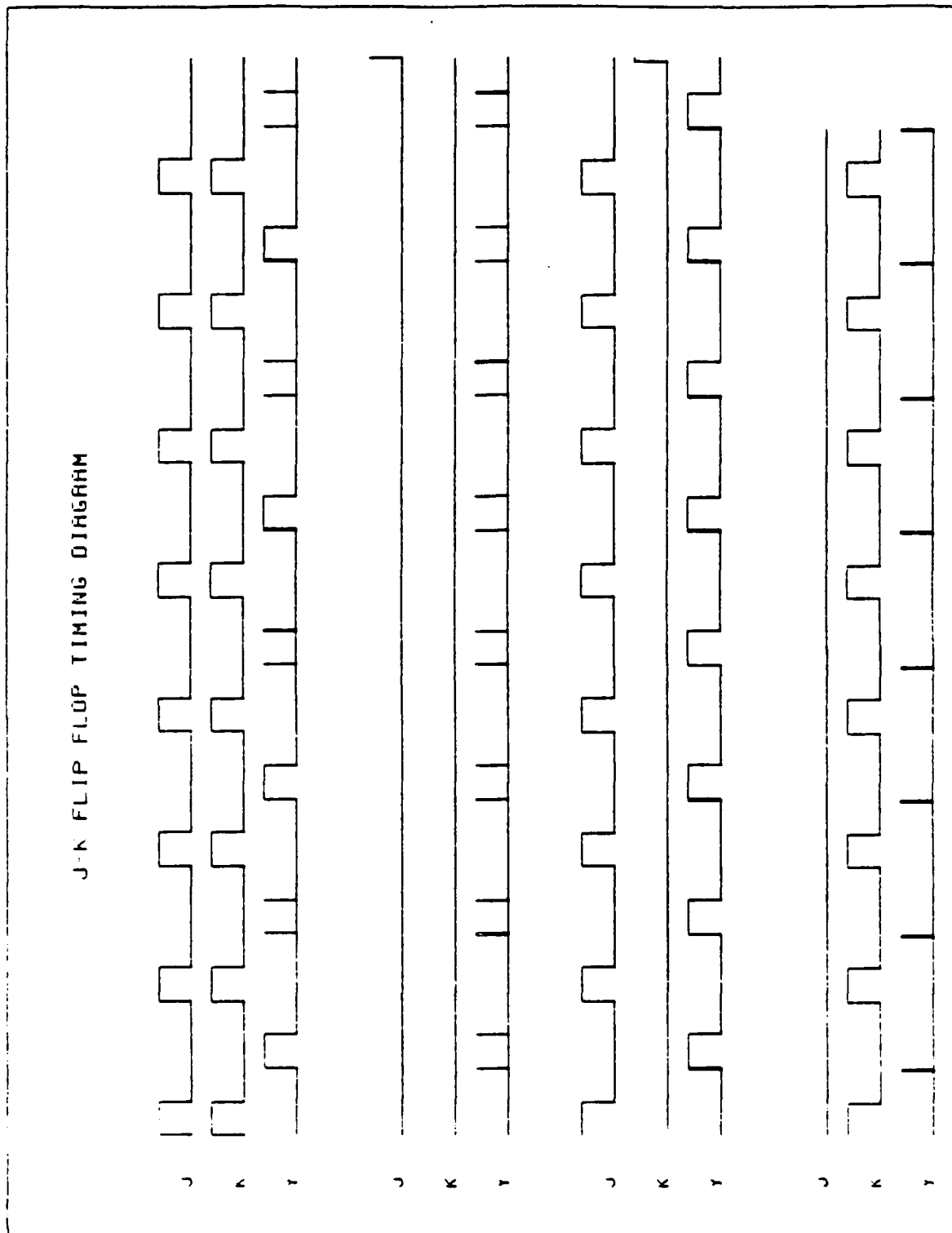


Figure 2-4. The Effects of a -0.1% Error in the Length of The Y Feedback Path

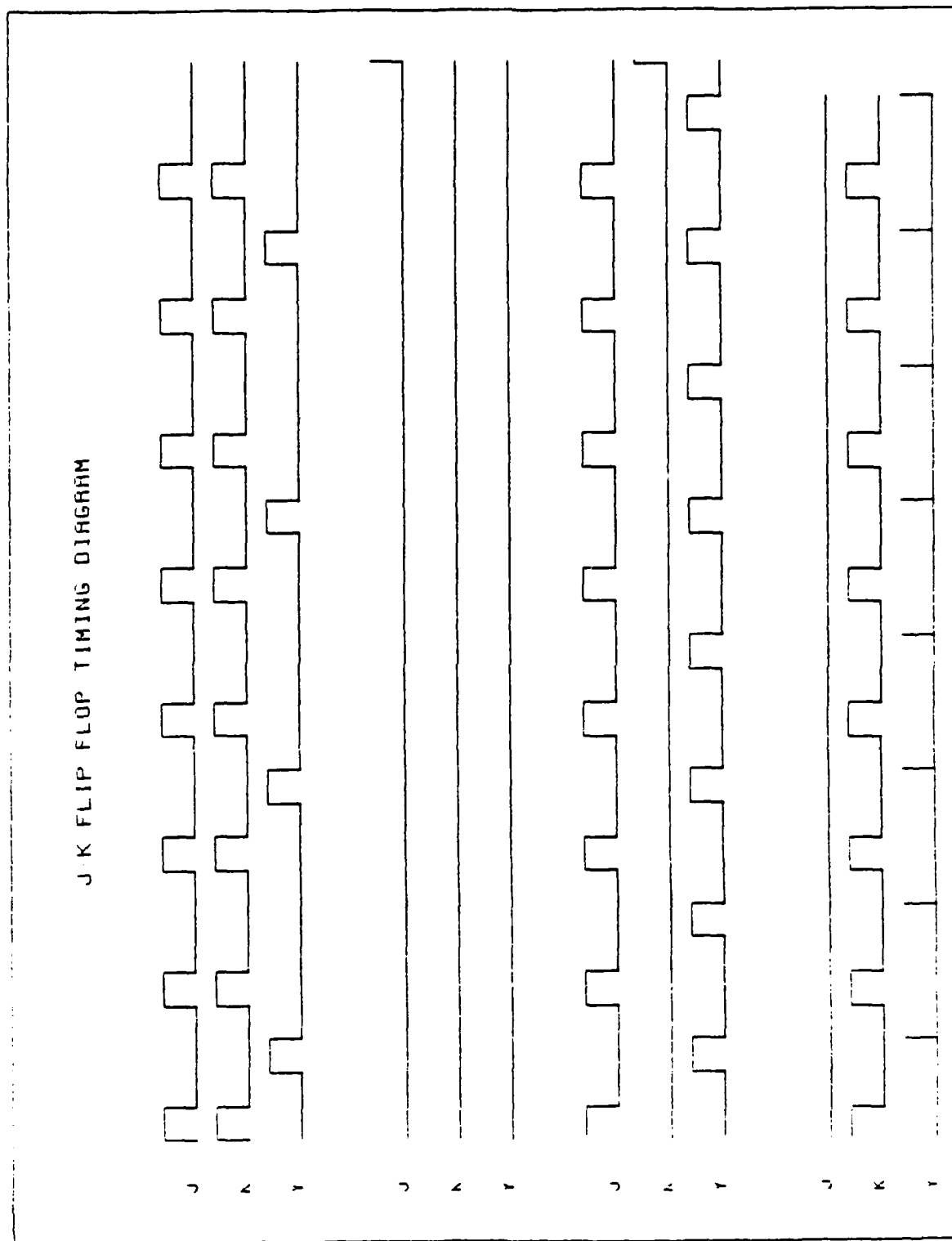


Figure 2-5. The Effects of a +0.1% Error in the Length of The Z Feedback Path

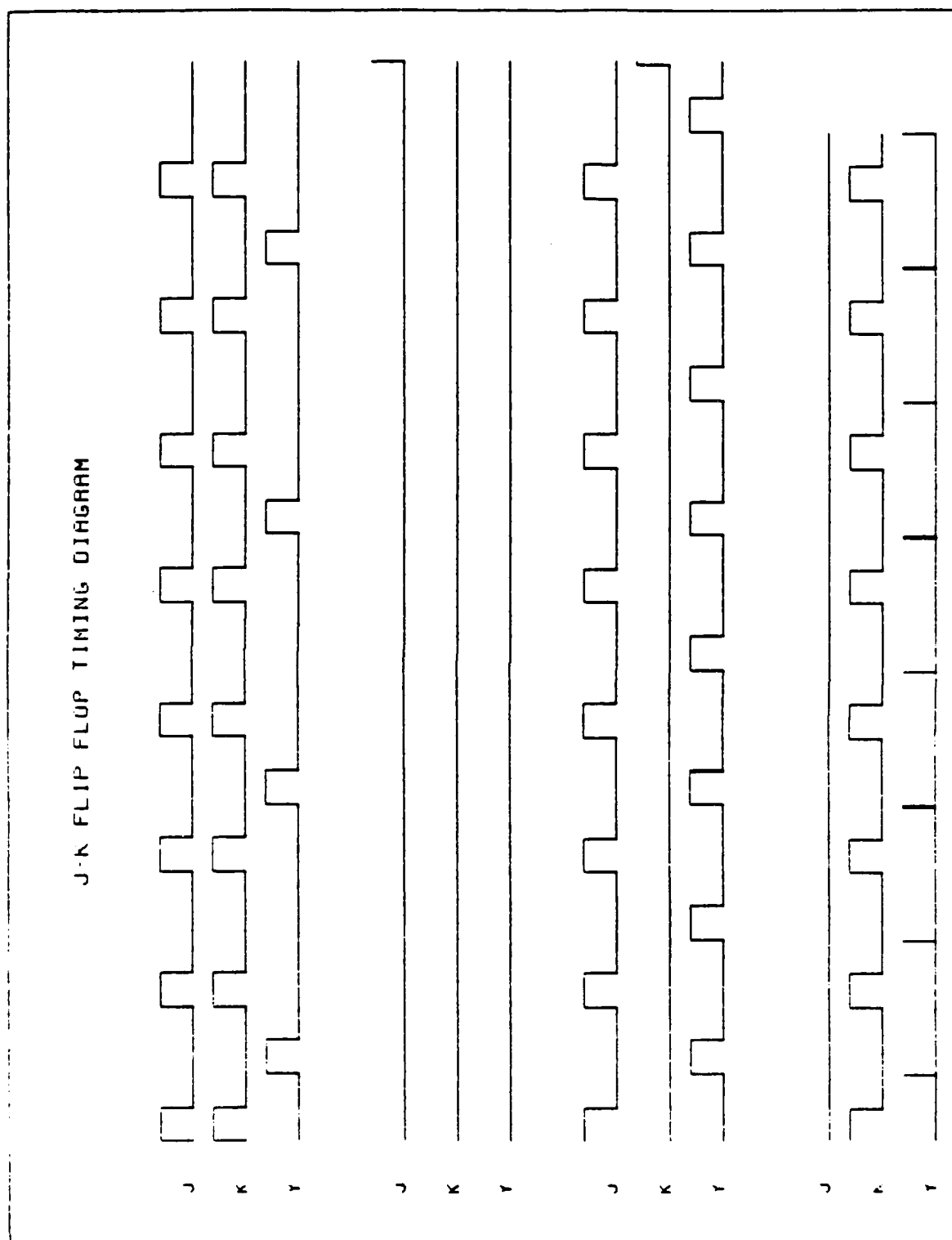


Figure 2-6. The Effects of a -0.1% Error in the Length of The Z Feedback Path

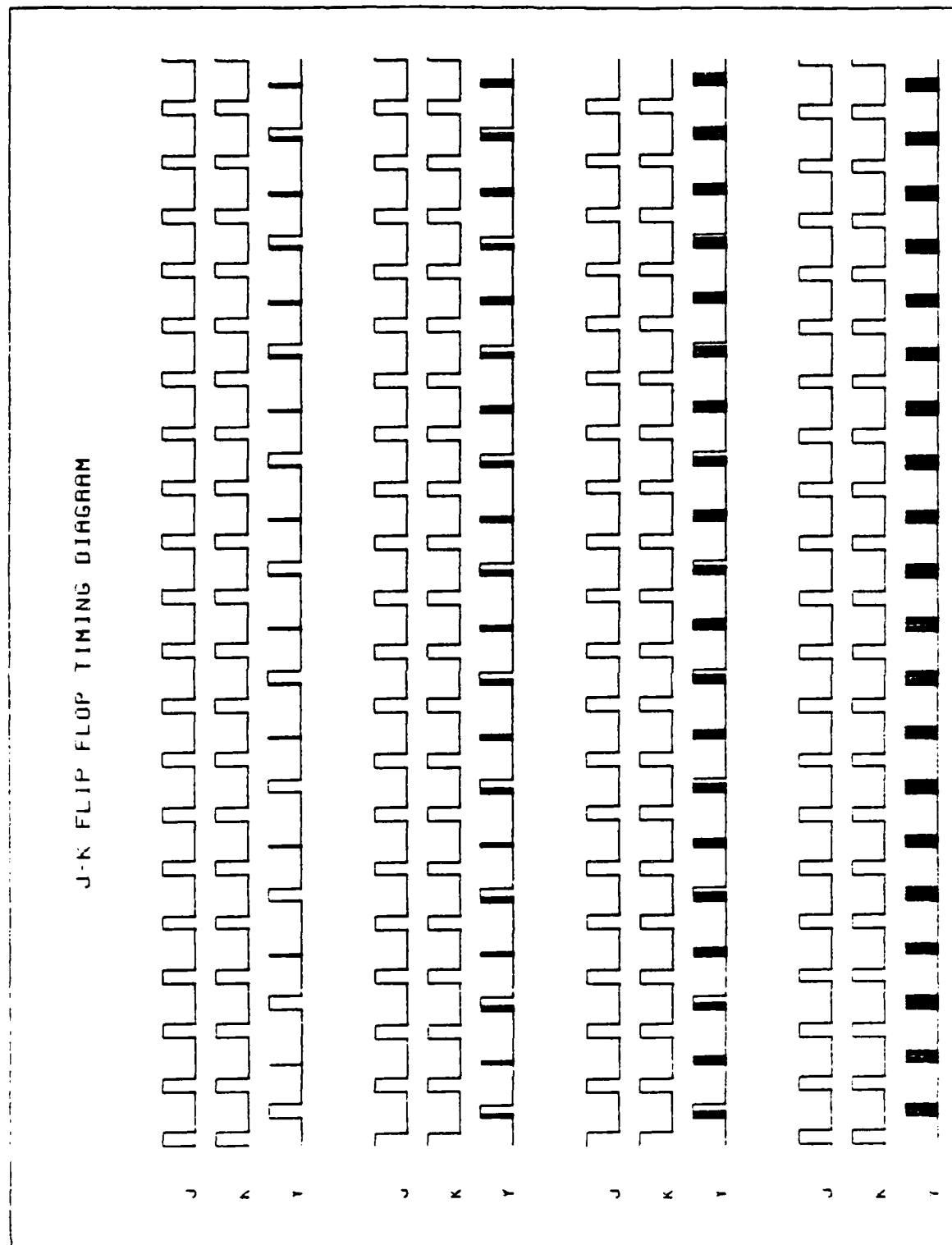


Figure 2-7. The Long-Term Effects of a +0.1% Error in The Length of The X Feedback Path

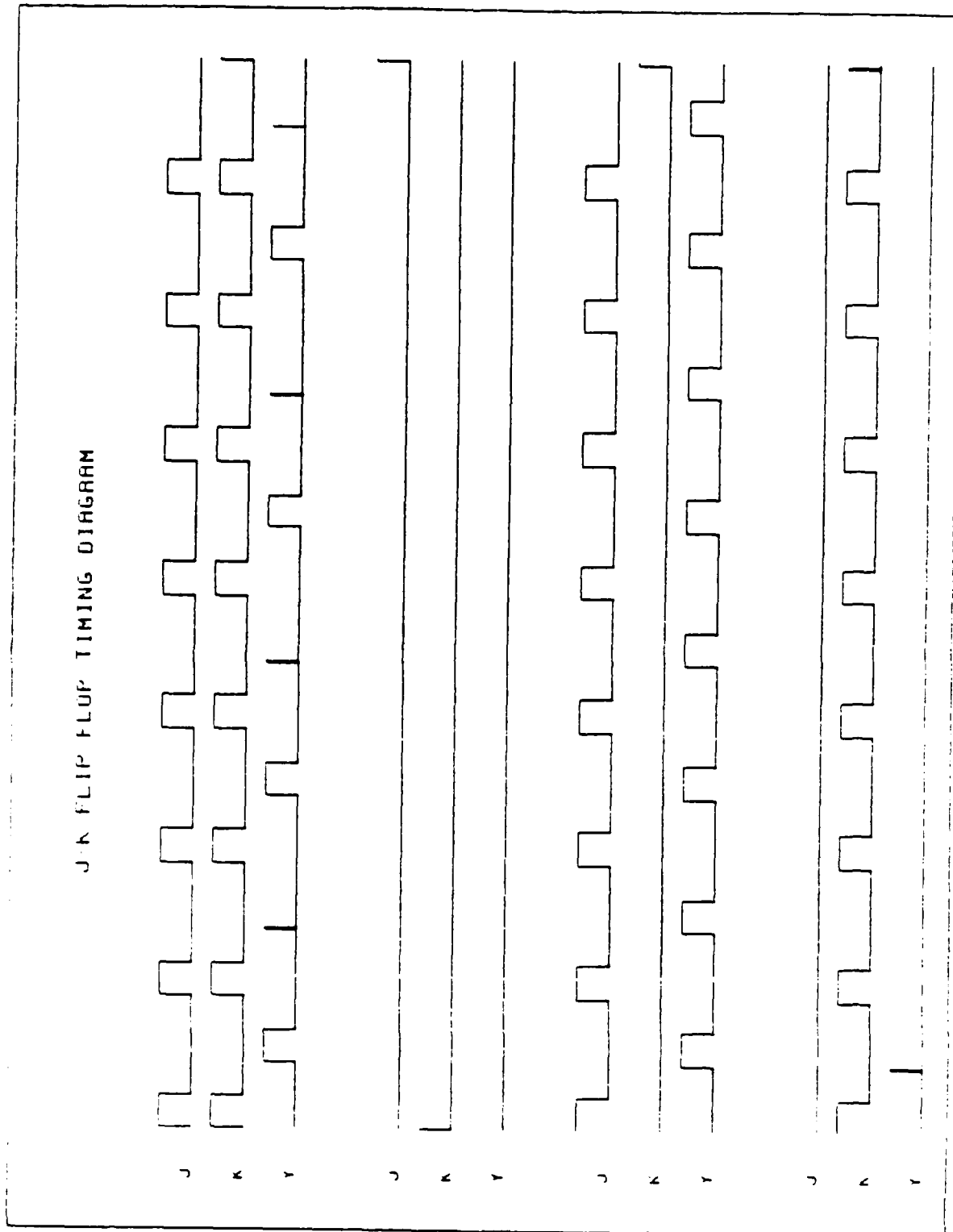


Figure 2-8. The Effects of the J Input Leading The K Input By 1%

## J-K FLIP FLOP TIMING DIAGRAM

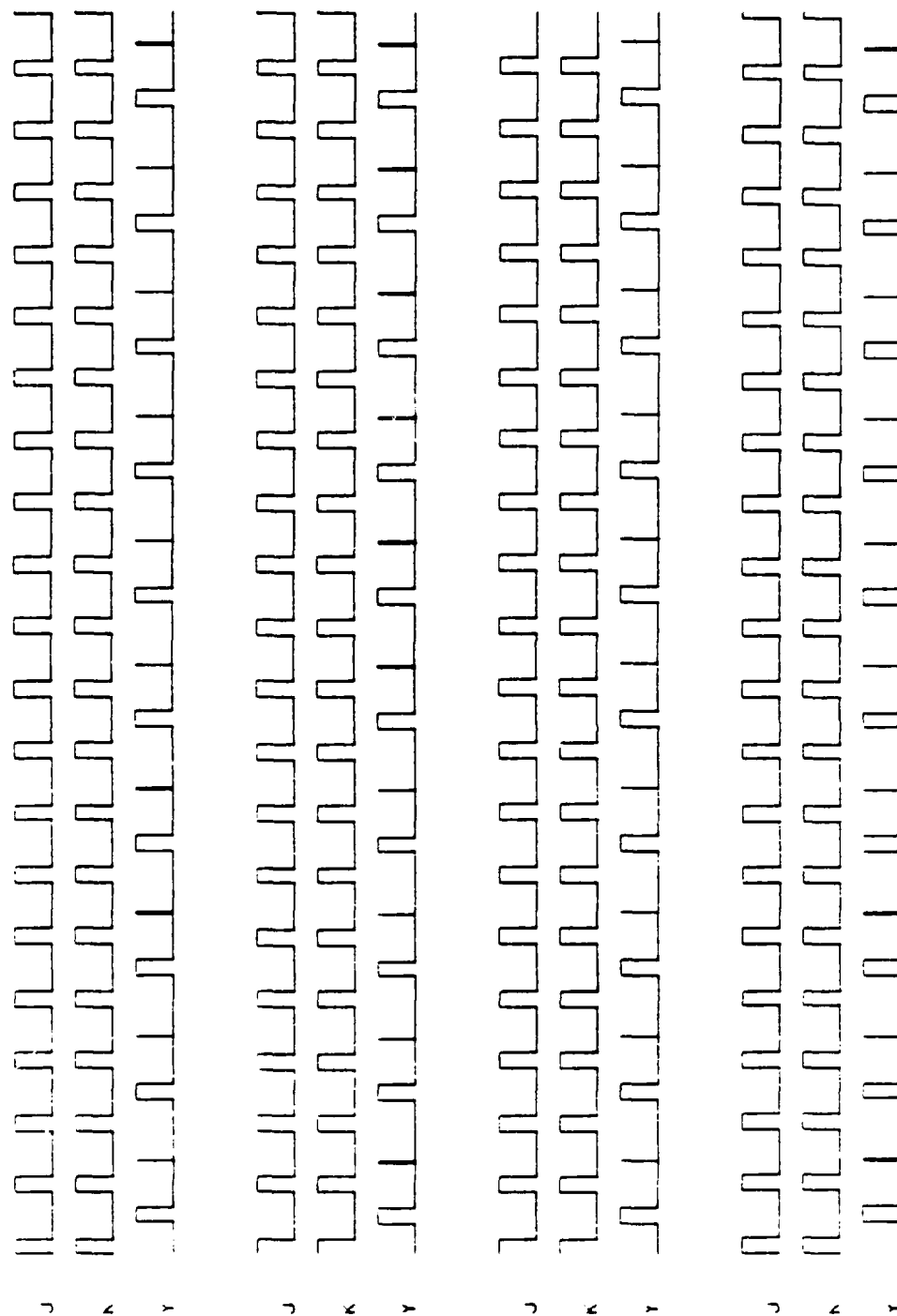


Figure 2-9. The Effects of the J Input Lagging The K Input By 1%

### 2.3 -- Continued.

cause a spreading of the pulse in time. This spreading could potentially introduce spurious signals at the leading and trailing edges of internally replicated pulses arriving synchronously with input pulses at the gain mechanism. This effect is portrayed in Figure 2-10, where the finite frequency response was simulated by the time correlation of the signals arriving at the gain mechanism with a Gaussian dispersion pulse. In this case, a filter  $1/e$  width of 0.2% of the pulse width was used.

Careful design with respect to the frequency response of the gain mechanism may, however, help to minimize spurious signals. In observing Figures 2-1 through 2-10, it is evident that the spurious signals appear as sharp spikes in the output. These spikes are of a higher frequency content and are more susceptible to suppression by the finite frequency response of the gain mechanism than the signal pulses. By choosing a pulse width, more closely matching the response time of the gain mechanism, better performance can be achieved, as measured by the ratio of the "true" signal response to the spurious signal response.

An example of this kind of improved performance is shown in Figure 2-11. This figure shows the effects of a Gaussian filter, of width equal to the input pulsewidth, on the spurious signals produced by a 1% error in the length of the X feedback path. In comparison with Figure 2-1, we see that the spurious signals have been effectively eliminated.

### 2.4 Effects of Mismatch Between Input Period and Round Trip Delay.

Another source of erroneous signals is that of a mismatch between the period of the input signals and twice the round trip delay. A rather extreme case is portrayed in Figure 2-12 where the input period is 5% shorter than the internal delay time with no filtering.

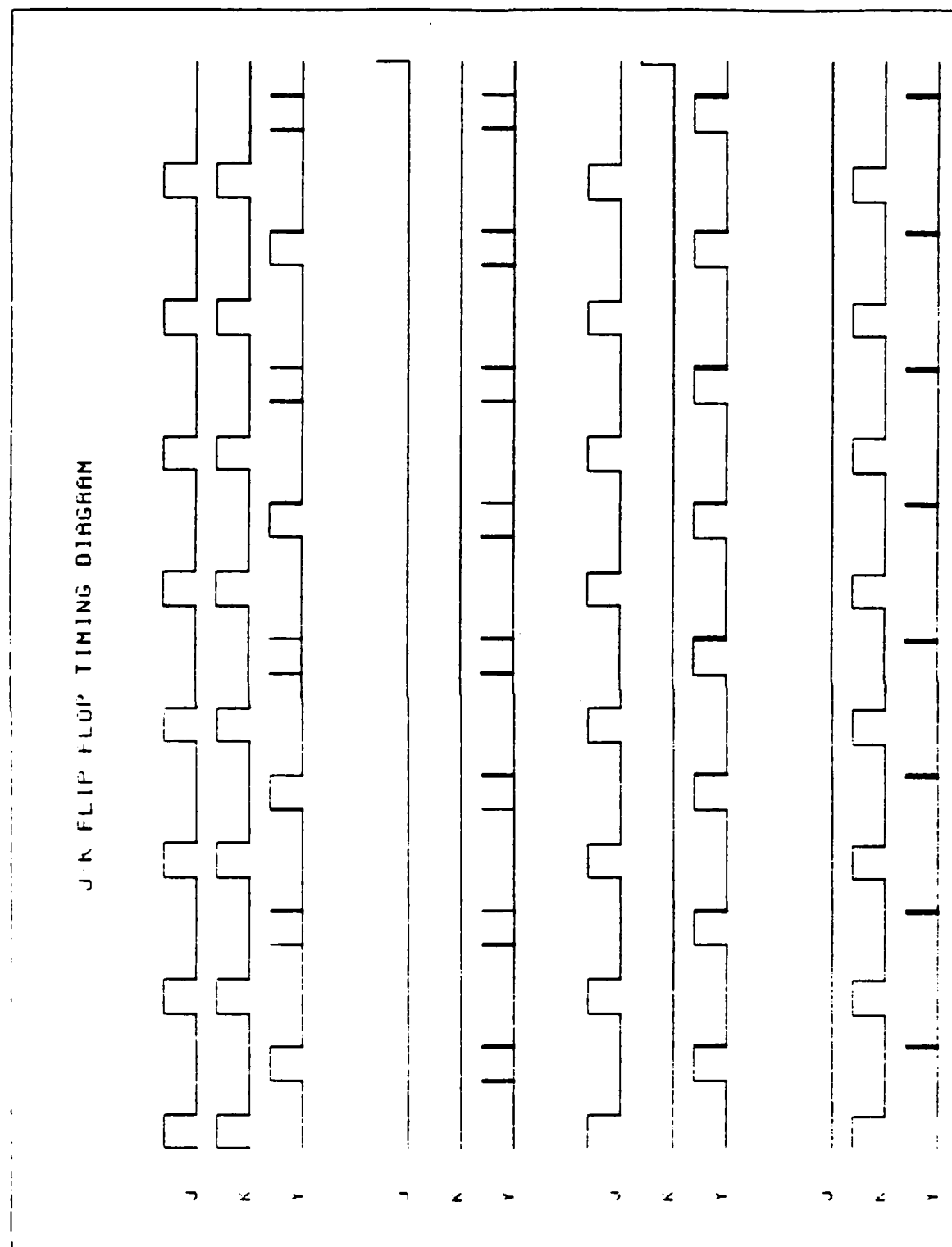


Figure 2-10. The Effects of Pulse Width Dispersion of 0.2%



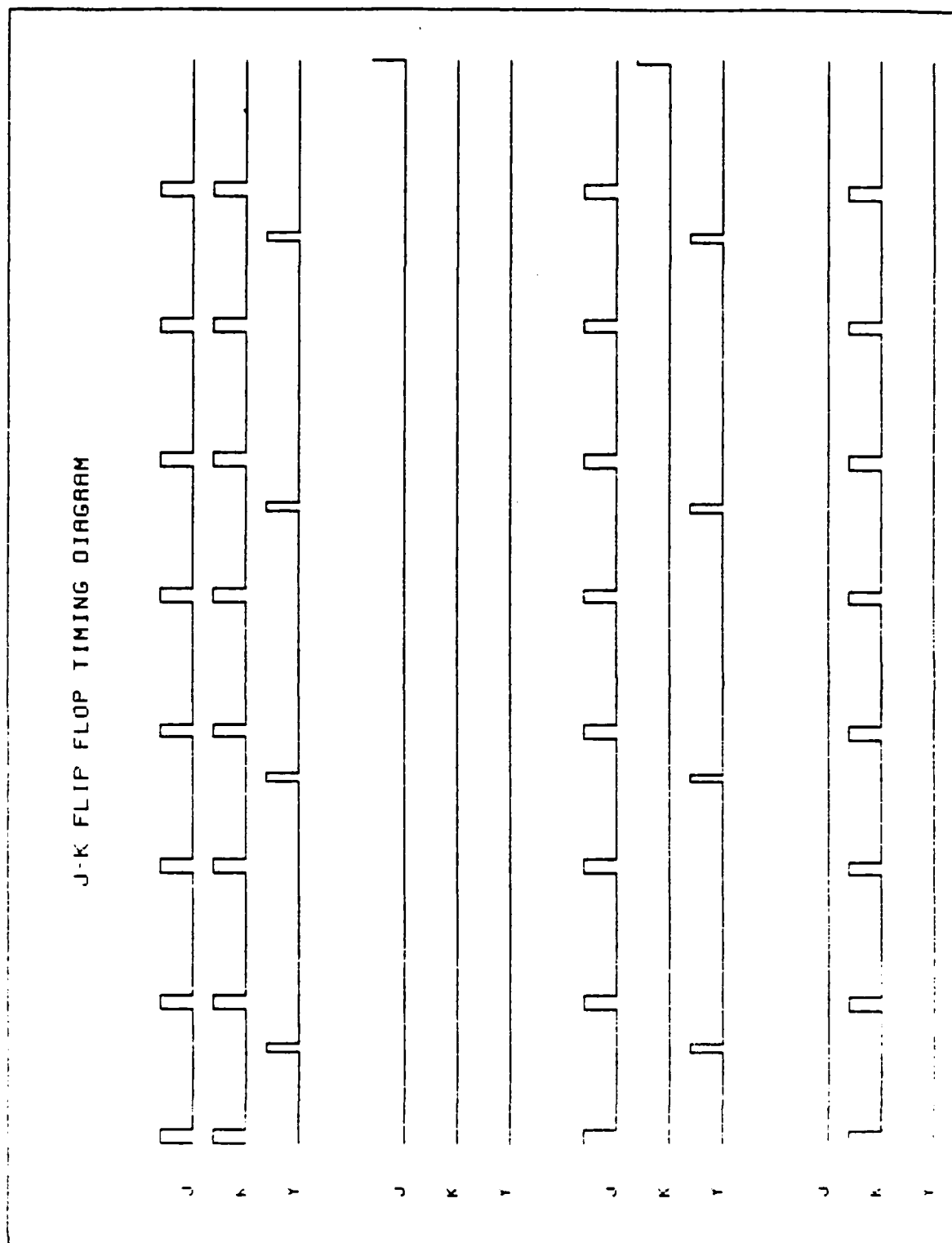


Figure 2-11. The Effects of Matched Filtering on Spurious Signals Introduced by a 1% Error in the X Path Length

## J-K FLIP FLOP TIMING DIAGRAM

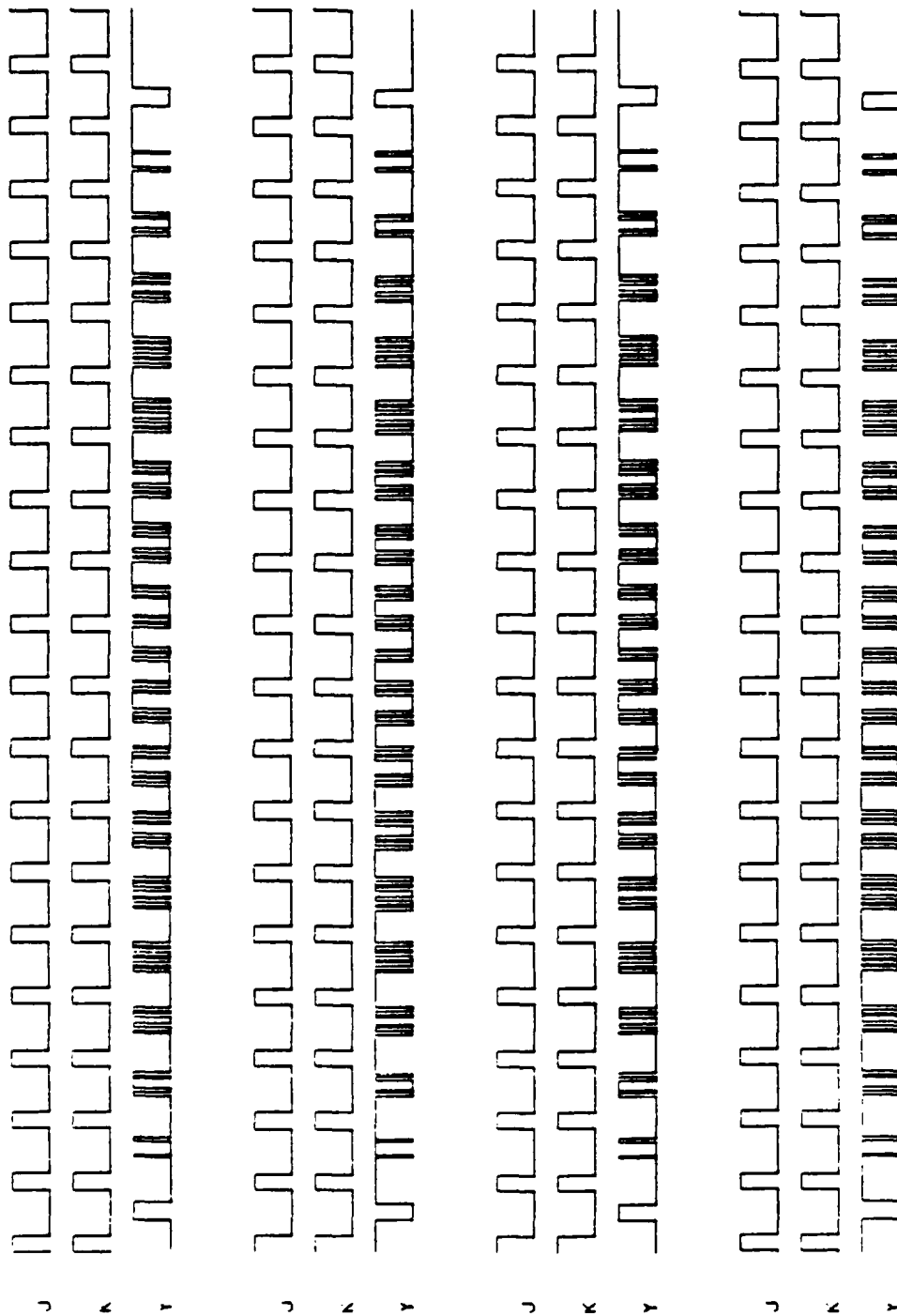


Figure 2-12. The Effects of the Input Clock Rate 5% Higher Than the Internal Clock Rate

2.4 -- Continued.

In general, spurious signals introduced by this pulse rate mismatch are similar to those introduced by path length errors and can be controlled to some degree by the filtering in the gain mechanism. However, in cases where the inputs contain long strings of "zeros" synchronization with the externally clocked signals will be lost.

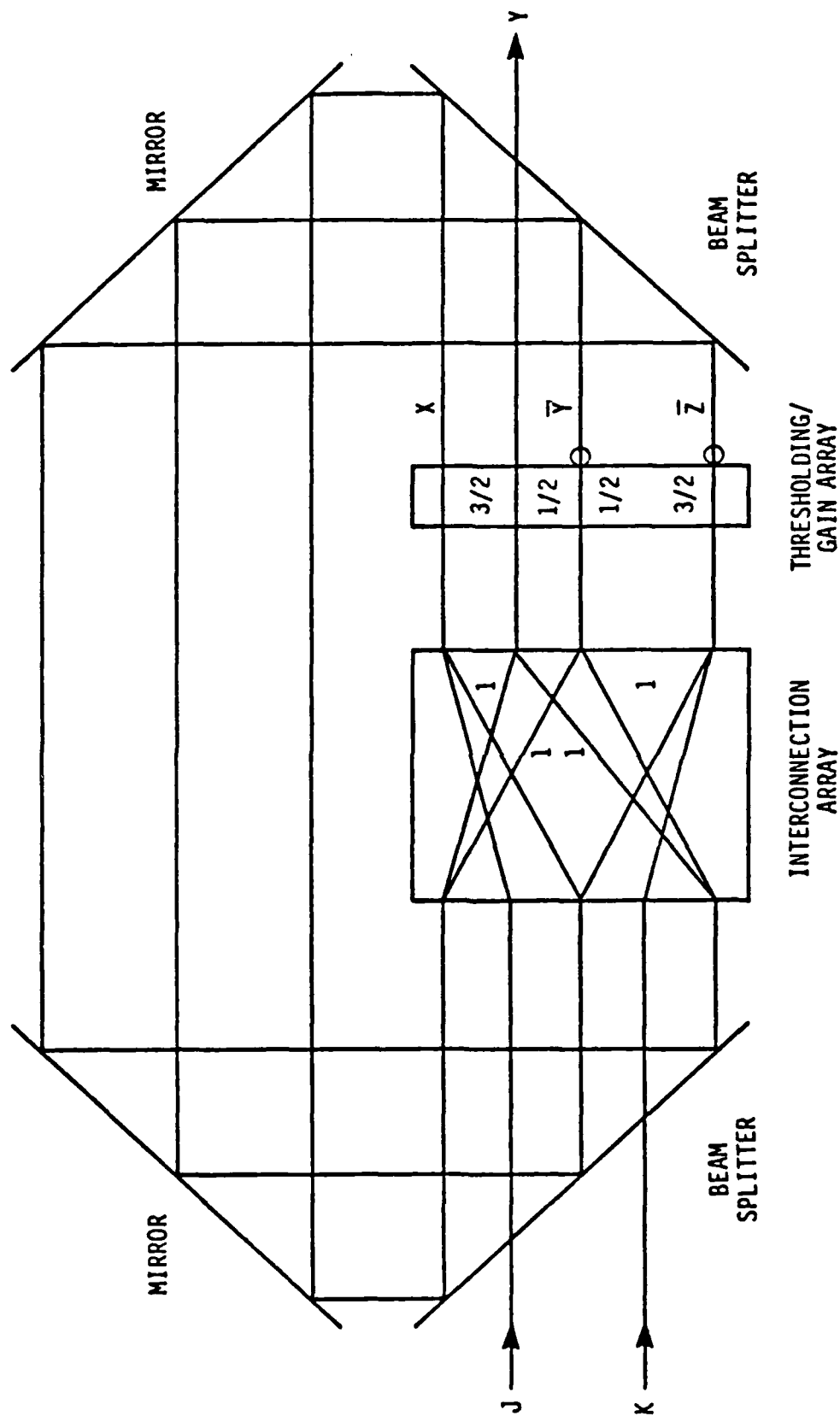
### 3. THRESHOLD AND WEIGHTING ANALYSIS.

This section will analyze the effects of random noise on the choice of the weights and thresholds in the optical J-K flip-flop. Two cases will be addressed. One case is the proposed implementation shown in Figure 1-1. This implementation utilizes coherent destructive interference to obtain negative weights. In this case, random phase noise complicates the analysis. Before proceeding to the analysis of the proposed case, we will first analyze another implementation using noncoherent light without interference (no negative weights) that is simpler to analyze and probably more readily realizable with current technology.

#### 3.1 Noncoherent Optical J-K Flip-Flop Analysis.

By noncoherent, it is meant that there exists no fixed phase relationship between the inputs of the flip-flop and the feedback paths. For destructive interference (and negative weights) a fixed temporal phase relationship between the input pulses of light and those being fed back is a necessary condition. In the noncoherent system, this condition is not satisfied and all weights must be positive. One method of enabling the proper operation of the flip-flop with all positive weights is to use inverting logic incorporated into the thresholding gain array. The necessary modifications to operate in an incoherent mode are shown in Figure 3-1.

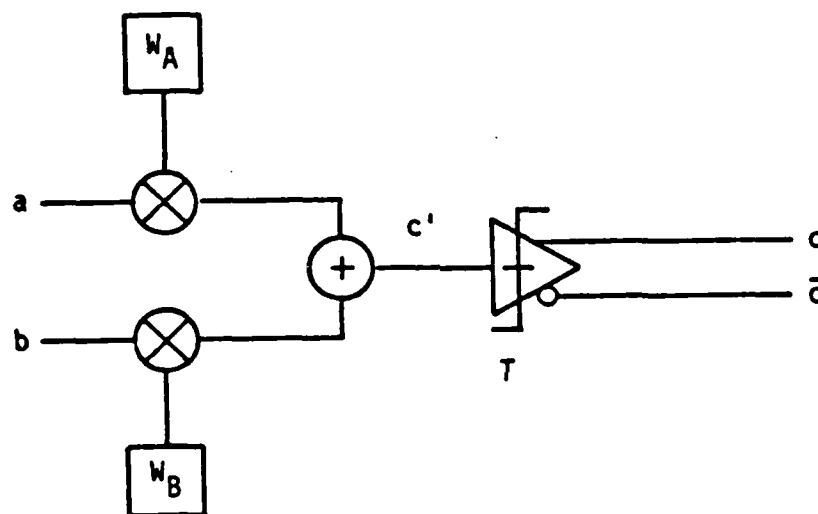
The problem of choosing the proper weights and thresholds in any implementation is one of choosing the weights and thresholds that will give the least chance of an erroneous output in the presence of noise at the inputs. In order to simplify the analysis, we note that each of the outputs of threshold gain array can be modeled independently using the simplified model shown in Figure 3-2. In this model, two inputs are weighted, summed, and compared with a threshold. The relationship between the output and the inputs is analogous to an "AND" or an "OR" gate depending on where the threshold is set.



PS-86005

PS-ER-5652-01

Figure 3-1. Noncoherent Implementation of Optical J-K Flip-Flop



PS-86006

Figure 3-2. Equivalent Model of Noncoherent Optical Interconnect and Threshold

### 3.1 -- Continued.

To further simplify the analysis, we will assume the inputs can be modeled as one of two equiprobable input levels as given in Equations (3-1) through (3-4).

$$a = A_0 + n_A \text{ (with probability } P_{A0} = 1/2) \quad (3-1)$$

$$a = A_1 + n_A \text{ (with probability } P_{A1} = 1/2) \quad (3-2)$$

$$b = B_0 + n_B \text{ (with probability } P_{B0} = 1/2) \quad (3-3)$$

$$b = B_1 + n_B \text{ (with probability } P_{B1} = 1/2) \quad (3-4)$$

In Equations (3-1) through (3-4),  $a$  and  $b$  are the inputs to the "gate",  $A$  and  $B$  are the input levels corresponding to a "mark" input or "zero" input depending on the subscript, and the  $n$ 's are noise terms. In this analysis the noise terms will be modeled as zero mean Gaussian distributed noise. The Gaussian probability density function is given by Equation (3-5).

$$f(n) = \frac{e^{-n^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \quad (3-5)$$

Where the  $\sigma^2$  is the variance of the noise. It should be noted that this model for the noise is only a rough approximation since it implies the possibility of negative inputs, however, we will use this model since it simplifies the analysis.

From the relations in Equations (3-1) through (3-4) the intermediate results shown in Equations (3-6) through (3-9) can be derived.

$$c' = c_{00} + n_c \text{ (with probability } P_{00} = 1/4) \quad c_{00} = A_0 \cdot W_A + B_0 \cdot W_B \quad (3-6)$$

$$c' = c_{01} + n_c \text{ (with probability } P_{01} = 1/4) \quad c_{01} = A_0 \cdot W_A + B_1 \cdot W_B \quad (3-7)$$

### 3.1 -- Continued.

$$c' = c_{10} + n_c \text{ (with probability } P_{10} = 1/4) \quad C_{10} = A_1 \cdot W_A + B_0 \cdot W_B \quad (3-8)$$

$$c' = c_{11} + n_c \text{ (with probability } P_{11} = 1/4) \quad C_{11} = A_1 \cdot W_A + B_1 \cdot W_B \quad (3-9)$$

Here  $c'$  is the weighted sum of the inputs (as shown in Figure 3-2). The noise terms are again Gaussianly distributed with variance equal to the sum of the variances of each of the input noise densities scaled by the appropriate weight.

The probability density functions for the four possibilities given in Equations (3-6) through (3-9) are given in Equations (3-10) through (3-13).

$$f_{00}(c') = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(c' - C_{00})^2/2\sigma^2} \quad (3-10)$$

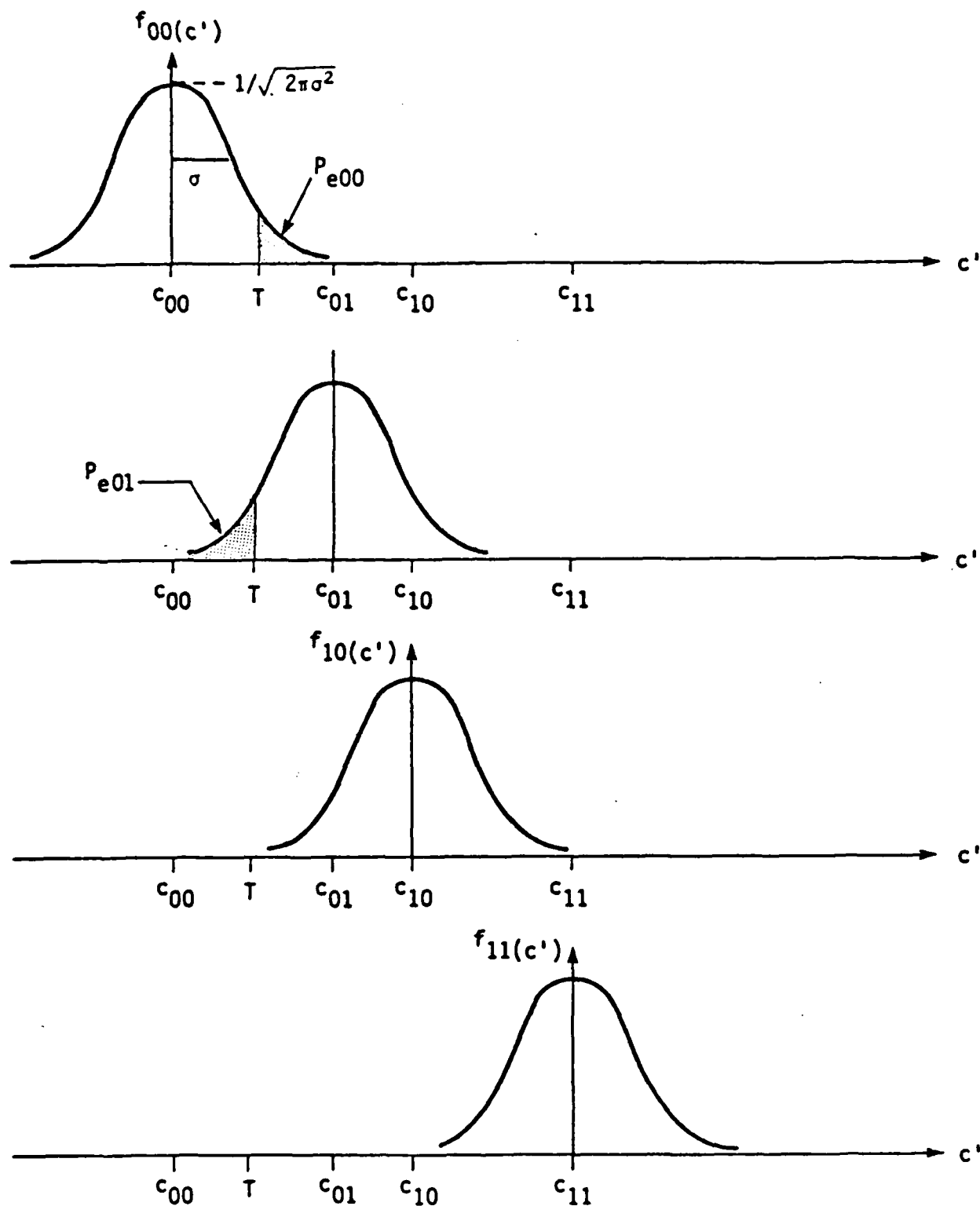
$$f_{01}(c') = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(c' - C_{01})^2/2\sigma^2} \quad (3-11)$$

$$f_{10}(c') = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(c' - C_{10})^2/2\sigma^2} \quad (3-12)$$

$$f_{11}(c') = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(c' - C_{11})^2/2\sigma^2} \quad (3-13)$$

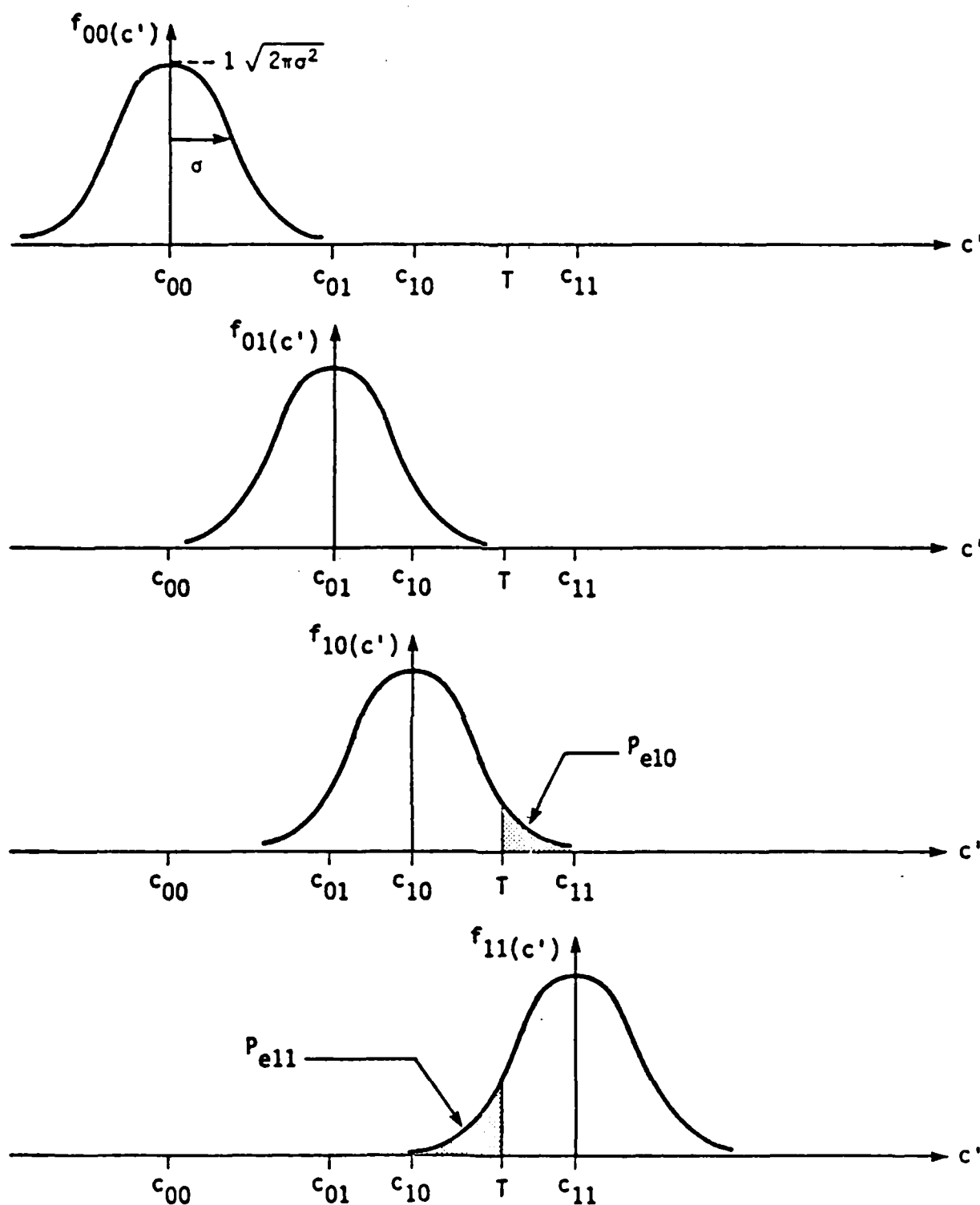
A graphical representation of these probability density functions is shown in Figure 3-3. Also shown in this figure is an example of the derivation of the probability of error for a threshold set for "OR" gate operation (the corresponding figure for "AND" gate operation is Figure 3-4). The total probability of error is the sum of the area under the top curve to the right of the threshold multiplied by its *a priori* probability (given in Equation (3-6)) and the areas of the lower three curves multiplied by their corresponding *a priori* probabilities. Expressions for the total error probability are given in Equations (3-14) and (3-15).





PS-86007

Figure 3-3. Probability Density Functions at Input to Threshold and Probability of Error Calculation For "OR" Operation



PS-86008

Figure 3-4. Probability Density Functions at Input to Threshold and Probability of Error Calculation For "AND" Operation

## 3.1 -- Continued.

$$P_e = P_{00} \cdot P_{e00} + P_{01} \cdot P_{e01} + P_{10} \cdot P_{e10} + P_{11} \cdot P_{e11} \quad (3-14)$$

$$P_e = P_{00} \int_T^{\infty} f_{00}(c') dc' + P_{01} \int_{-\infty}^T f_{01}(c') dc' \\ + P_{10} \int_{-\infty}^T f_{10}(c') dc' + P_{11} \int_{-\infty}^T f_{11}(c') dc' \quad (3-15)$$

The problem of selecting a threshold then becomes one of minimizing the expression given in Equation (3-15). Mathematically, this is done by taking the derivative of the probability of error with respect to the threshold value and setting the resulting expression to zero. The result of these operations is given in Equation (3-16).

$$\begin{aligned} & -P_{00} e^{-(T - C_{00})^2 / 2\sigma^2} + P_{01} e^{-(C_{01} - T)^2 / 2\sigma^2} + P_{10} e^{-(C_{10} - T)^2 / 2\sigma^2} \\ & + P_{11} e^{-(C_{11} - T)^2 / 2\sigma^2} = 0 \end{aligned}$$

Equation (3-16) not only yields a direct method of selecting the optimum threshold, but also gives some insight into the problem of choosing the optimum weight. The expression for the total probability of error in Equation (3-15), can be minimized with respect to the weights by minimizing the arguments of the exponentials in Equation (3-16). It seems the choice of weights is rather arbitrary since both the noise and the signal are weighted equally. However, since additional noise may be introduced after the weighting (particularly in the detection process) the weights should be chosen to maximize the difference between the mark and zero. Given that the weights are passive, the choices given in Equations (3-17) and (3-18) are probably the best.

### 3.1 -- Continued.

$$W_A = 1 \quad (3-17)$$

$$W_B = 1 \quad (3-18)$$

Equations (3-17) and (3-18) assume that the input amplitudes are equal. Should this not be the case, the weight should be chosen so as to equalize the inputs. For example, in the case where the b input is of greater amplitude than the a input, the weights should be chosen as given in Equations (3-19) and (3-20).

$$W_A = 1 \quad (3-19)$$

$$W_B = A_1/B_1 \quad (3-20)$$

With the weights given in Equations (3-17) and (3-18), the thresholds corresponding to "OR" and "AND" operation are given in Equations (3-21) and (3-22) respectively (scaled to unit amplitude inputs).

$$T_{OR} = 1/2 \text{ (Assuming } A_0 = B_0 = 0 \text{ and } A_1 = B_1 = 1) \quad (3-21)$$

$$T_{AND} = 3/2 \text{ (Assuming } A_0 = B_0 = 0 \text{ and } A_1 = B_1 = 1) \quad (3-22)$$

### 3.2 Coherent Optical J-K Flip-Flop Analysis.

As noted, the primary difference between the noncoherent and coherent implementations of the optical J-K flip-flop, is that the noncoherent implementation does not require the use of inverted logic signals. Instead the inverted logic is achieved by the use of negative weighting in the weighted interconnect array. To achieve the negative weighting, the holographic interconnect array shifts the phase of one signal with respect to a second so as to produce destructive interference at the thresholding gain array.

3.2      -- Continued.

To permit destructive interference, it is implicit that each of the signals arriving at the holographic interconnect must maintain a fixed phase relationship with respect to each other. The practical implications of this constraint are fairly imposing. Some of these implications are as follows:

- A. The J and K inputs must maintain a constant phase between successive pulses. In practice this is not easy to achieve. This means that some forms of modulation of the inputs (e.g. direct modulation) are not suitable. In addition, it seems that both inputs should be derived from the same source. And finally, the coherence time of the source laser must be much longer than the length of time that any signal remains inside the flip-flop.
- B. The lengths of the feedback paths must be accurately controlled to fractions of a wavelength. In order to provide this accuracy the flip-flop must be immune to any environmental sources of vibration and/or temperature dependent path length errors. Again, these problems are difficult to eliminate.
- C. The thresholding gain mechanism must not only have the desirable thresholding characteristics, but must also be able to both detect the relative phase of two signals and produce an output signal with a fixed phase relative to one of the two signals. In practice, there are devices capable of detecting the relative phase (interference) of two signals (e.g. square law detector); and there are devices capable of replicating the phase of an input signal (e.g. laser amplifier); but there is no obvious simple mechanism for performing both these functions in addition to thresholding.

### 3.2 -- Continued.

These are interesting problems for research and development; and in order to proceed with the analysis, we will assume that they can be resolved. Thus, we will assume that the operation of the coherent optical J-K flip-flop is not corrupted by any systematic errors, but only by random noise components similar to those discussed in reference to the noncoherent case.

The simplified model of the weighted interconnect array and thresholding gain medium for the coherent case is shown in Figure 3-5. The differences between this model and that of the noncoherent case (Figure 3-1) are: first, the thresholding gain mechanism does not output an inverted signal; and second, the signals as well as the weights are complex (i.e. they possess both magnitude and phase).

For convenience, we will represent the signals as phasors as shown in Equations (3-23) and (3-24).

$$a = A + n_A$$

$$A = |A| e^{i\omega t} e^{i\theta_A} \quad (3-23)$$

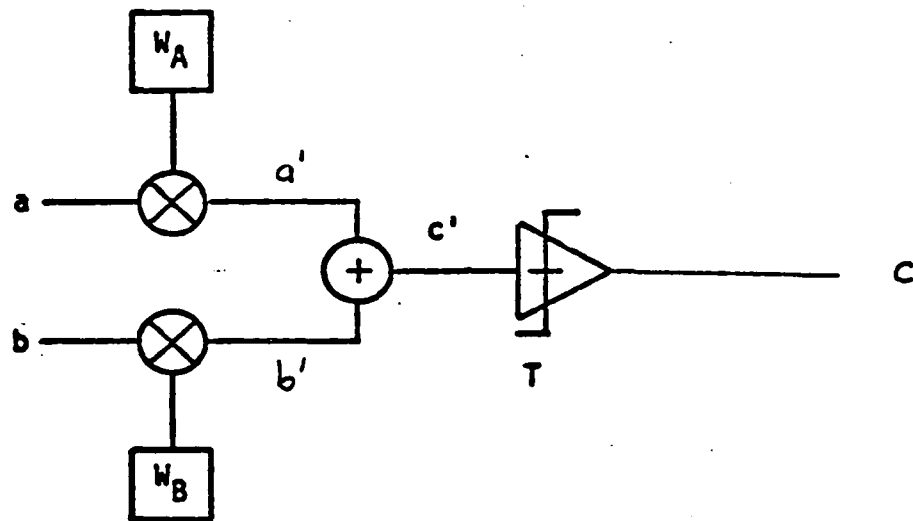
$$n_A = |n_A| e^{i\omega\phi_A}$$

$$b = B + n_B$$

$$B = |B| e^{i\omega t} e^{i\theta_B} \quad (3-24)$$

$$n_B = |n_B| e^{i\omega\phi_B}$$

In this case, the signal phasors (A and B) can be modulated by either phase or amplitude; however, for this analysis, we will assume that the signals are modulated using an on-off modulation with equiprobable states. In addition, we



PS-86006

Figure 3-5. Equivalent Model of Coherent Optical Interconnect and Threshold

### 3.2 -- Continued.

will assume that the carrier frequencies are perfectly matched, and thus neglect the time variation for the remainder of the analysis. Finally, we will assume that the real and imaginary parts of the noise terms can be represented as independent zero mean Gaussianly random variables; or equivalently that the magnitude of the noise term is Rayleigh distributed and the phase angle is linearly distributed between 0 and  $2(\pi)$ .

As stated, the weights for the coherent case are complex as given in Equations (3-25) and (3-26).

$$w_A = |w_A| e^{i\delta_A} \quad (3-25)$$

$$w_B = |w_B| e^{i\delta_B} \quad (3-26)$$

The effects of weighting on the input signals are shown using the phasor representation for example in Figure 3-6. This figure shows the effects on an arbitrary signal with noise, of a weighting of unit magnitude and a  $180^\circ$  phase shift. The results of the weighting of the input signals are given in Equations (3-27) and (3-28).

$$a' = A' + n_A'$$

$$A' = |A'| e^{i\theta_A'} = |A| |w_A| e^{i(\theta_A + \delta_A)} \quad (3-27)$$

$$n_A' = |n_A| e^{i\theta_A} = |n_A| |w_A| e^{i(\phi_A + \delta_A)}$$



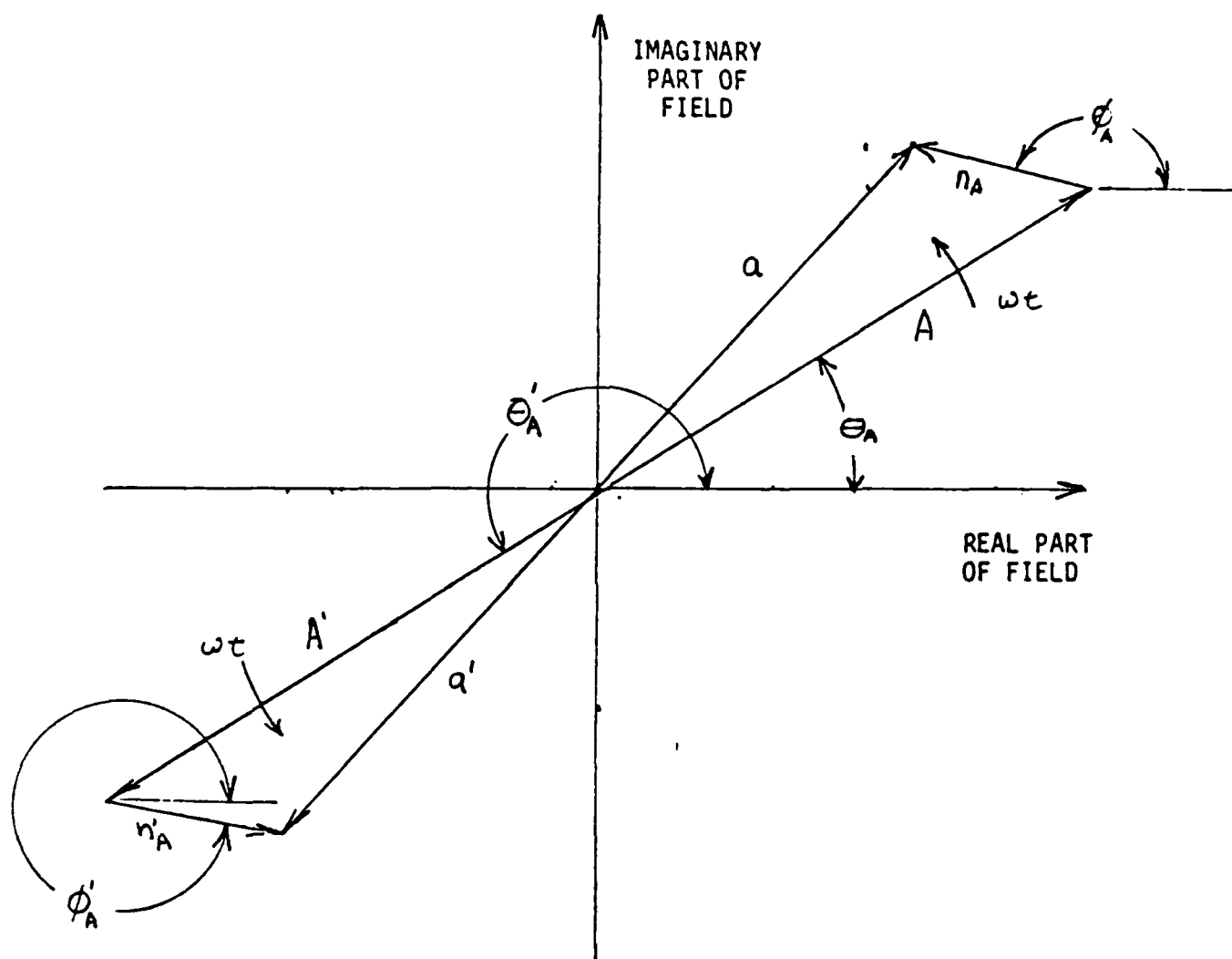


Figure 3-6. Phasor Representation of Complex Weighting of Input Signal

This figure shows the effects of complex weighting (magnitude and phase) on an arbitrary input signal with a noise component. In this example, the holographic weighting shifts the phase angles by  $180^\circ$  and scales the magnitudes by unity.

## 3.2 -- Continued.

$$b' = B' + n_B'$$

$$B' = |B'| e^{i\theta_B'} = |B| |w_B| e^{i(\theta_B + \delta_B)} \quad (3-28)$$

$$n_B' = |n_B| e^{i\theta_B'} = |n_B| |w_B| e^{i(\phi_B + \delta_B)}$$

At this point the signals are summed. This summing can be represented as the vector sum of the two weighted phasors as shown for example in Figure 3-7. The threshold operation will be based on the magnitude (squared) of the vector sum.

To determine the vector sum, it is convenient to change coordinate systems using the coordinate transformations given in Equations (3-29) through (3-32).

$$A' = X_A + i Y_A$$

$$X_A = |A'| \cos \theta_A' \quad (3-29)$$

$$Y_A = |A'| \sin \theta_A'$$

$$n_A' = X_A + i Y_A$$

$$X_A = |n_A'| \cos \phi_A' \quad (3-30)$$

$$Y_A = |n_A'| \sin \phi_A'$$

$$B' = X_B + i Y_B$$

$$X_B = |B'| \cos \theta_B' \quad (3-31)$$

$$Y_B = |B'| \sin \theta_B'$$

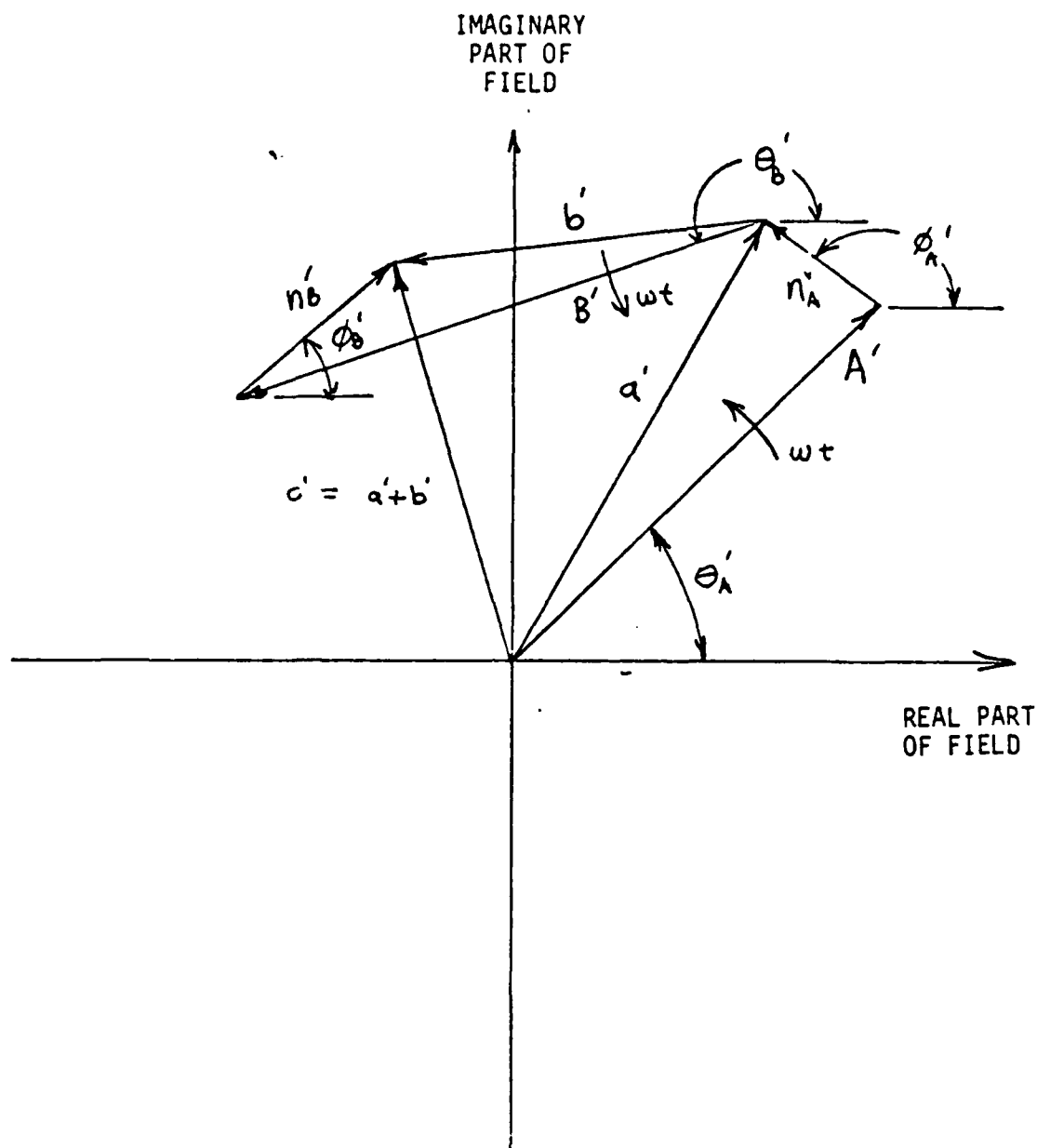


Figure 3-7. Phasor Representation of the Sum of Two Arbitrary Signals With Noise Components

### 3.2 -- Continued.

$$n_B' = x_B + i y_B$$

$$x_B = |n_B'| \cos \phi_B' \quad (3-32)$$

$$y_B = |n_B'| \sin \phi_B'$$

Using these transformations, the vector sum can be expressed as given in Equation (3-33).

$$c' = x_c + i y_c$$

$$x_c = x_A + x_B \quad (3-33)$$

$$y_c = y_A + y_B$$

$$x_c = x_A + x_B$$

$$y_c = y_A + y_B$$

Here, the real and imaginary components of the noise ( $x_c$  and  $y_c$ ) are still zero mean, Gaussian random variable with variance equal to the sum of the component variances (weighted by magnitude of the corresponding weight).

The magnitude of the sum is given by Equation (3-34).

$$|c'|^2 = c' \cdot c'^*$$

$$|c'|^2 = (x_c + c_c)^2 + (y_c + y_c)^2 \quad (3-34)$$

The magnitude squared given in Equation (3-34) is a random variable with second order non-central chi-squared density as given in the density function, Equation (3-35).

### 3.2 -- Continued.

$$f(|c'|^2) = \frac{1}{2\sigma} e^{-(|c'|^2 - C^2)/2\sigma^2} \quad (3-35)$$

In this equation ( $\sigma^2$ ) is the noise variance and  $C^2$  is the mean square value, as a function of the weights and input amplitudes. A graphical example of chi-squared density function is given in Figure 3-8.

At this point expressions for the probability of error and optimum threshold can be derived analogous to those for the noncoherent case given in Equations (3-16) and (3-17). But rather, preferred weights and thresholds will be given based on more general arguments similar to those used at the end of Section 3.1.

The optical J-K flip-flop performs three logic operations. The equivalent logic element, truth table, optimum weight selection and optimum threshold selection are shown for each of the three cases in Figures 3-9 through 3-11. The optimum weights were selected by maximizing the separation between the peaks of the density functions; and the optimum thresholds were selected by minimizing the shaded areas under the density functions.

An important observation to be made is that the errors are much less probable for the "AND" gate shown in Figure 3-11 than for either of the other gates shown in Figures 3-9 and 3-10. Two other "preferred" gates are shown in Figures 3-12 and 3-13. In general, use of combinations of the "preferred" gates shown in Figures 3-11 through 3-13 will help minimize errors. For the J-K flip-flop, however, exclusive use of these gates would require many more gates for the same operation.

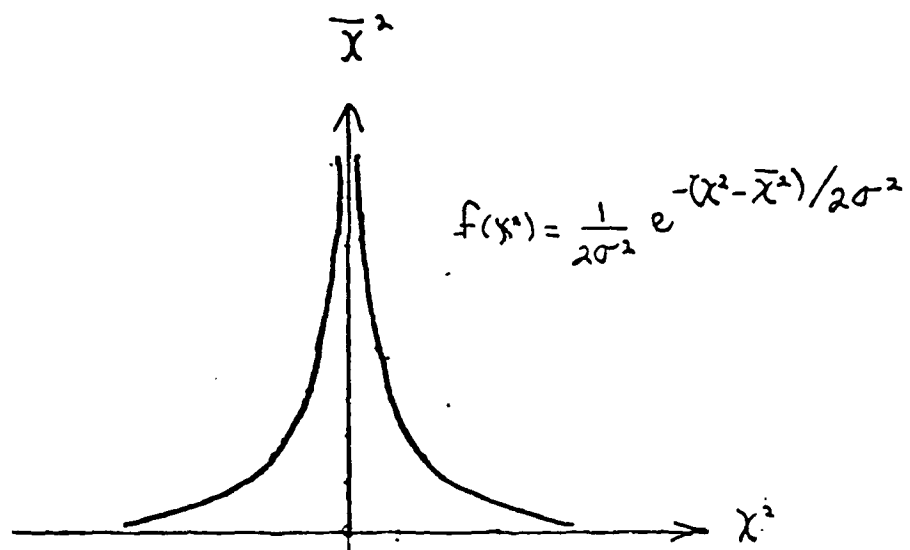
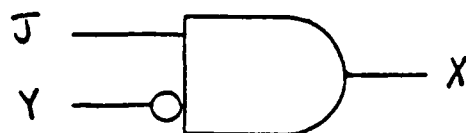


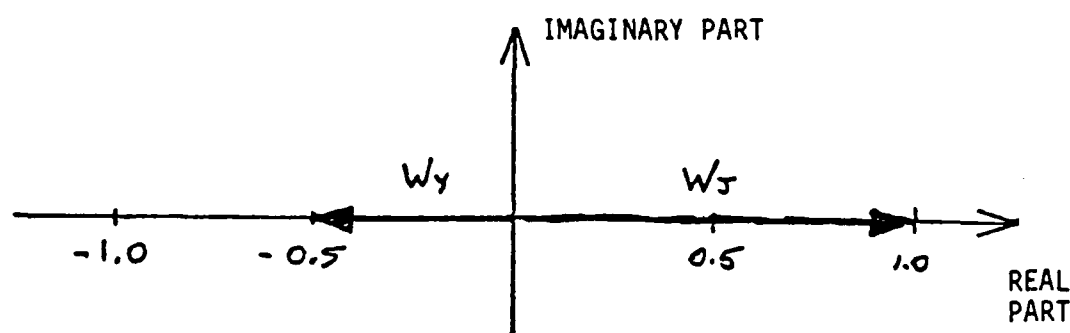
Figure 3-8. Chi-Squared Density Function



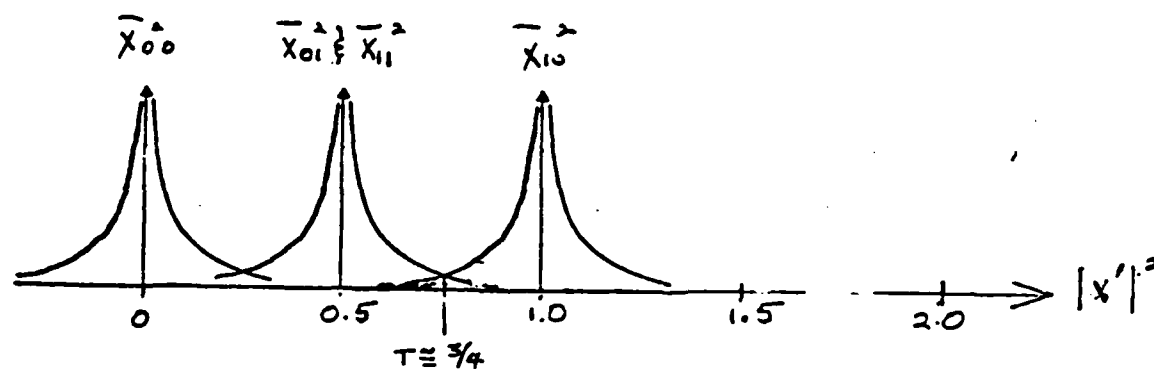
a. Equivalent Logic Gate

J	Y	X
0	0	0
0	1	0
1	0	1
1	1	0

b. Truth Table

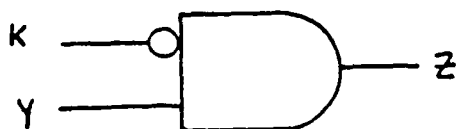


c. Complex Weights



d. Density Functions and Threshold Selection

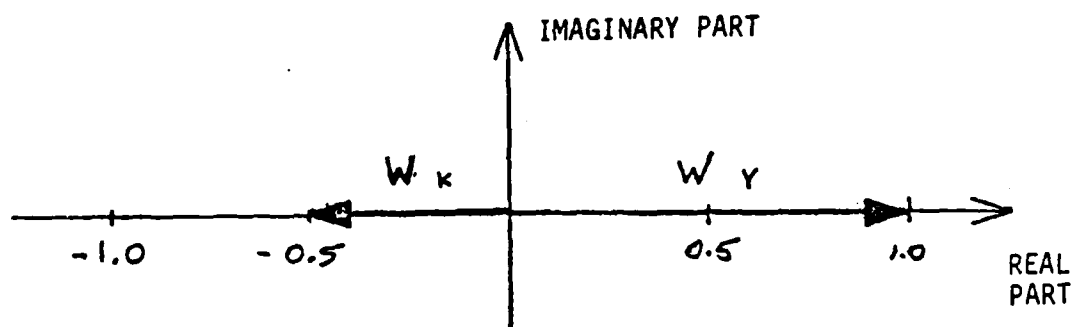
Figure 3-9. Optimum Weights and Threshold For X Logic Operation



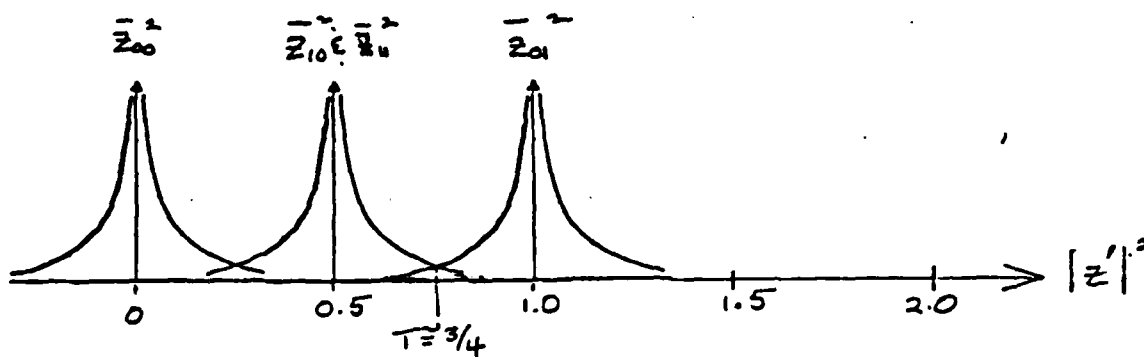
a. Equivalent Logic Gate

K	Y	Z
0	0	0
0	1	1
1	0	0
1	1	0

b. Truth Table



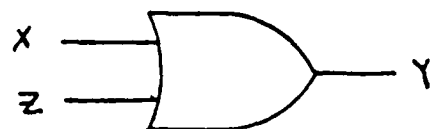
c. Complex Weights



d. Density Functions and Threshold Selection

Figure 3-10. Optimum Weights and Threshold For Z Logic Operation

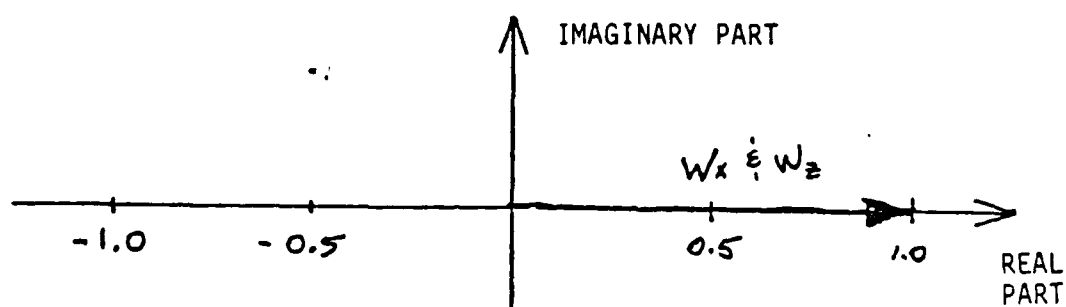




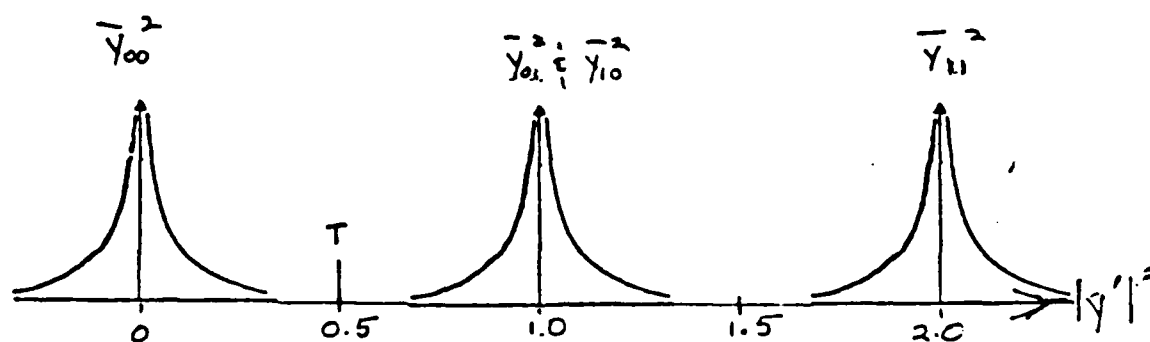
a. Equivalent Logic Gate

X	Z	Y
0	0	0
0	1	1
1	0	1
1	1	1

b. Truth Table

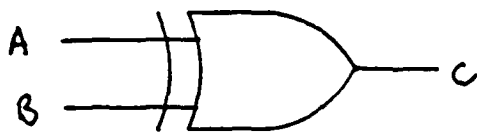


c. Complex Weights



d. Density Functions and Threshold Selection

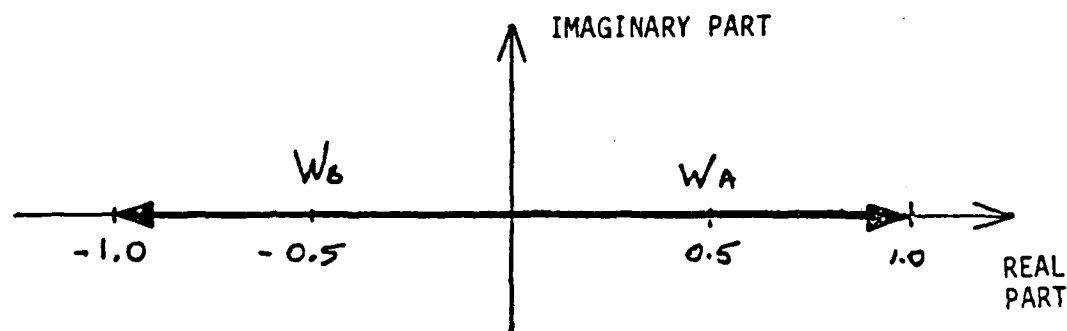
Figure 3-11. Optimum Weights and Threshold For "OR" Logic Operation



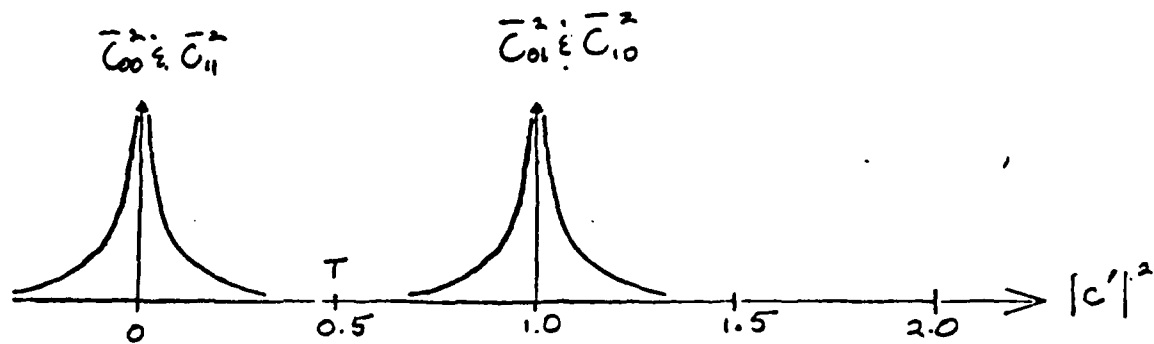
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

a. Equivalent Logic Gate

b. Truth Table

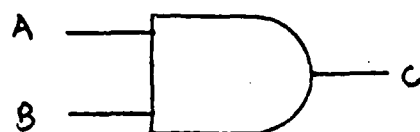


c. Complex Weights



d. Density Functions and Threshold Selection

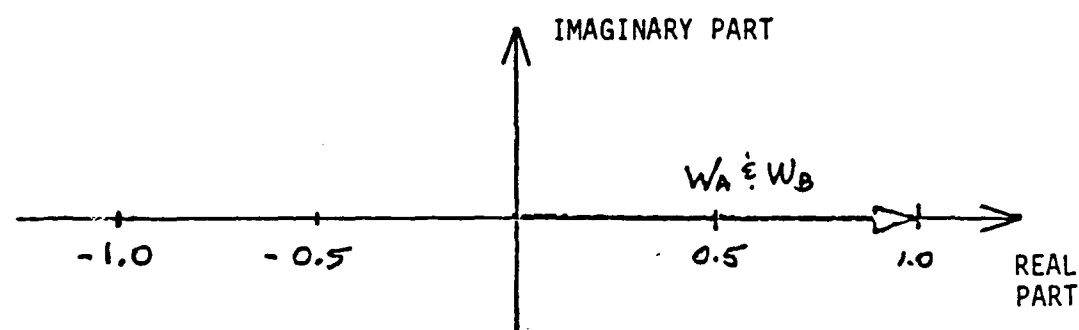
Figure 3-12. Optimum Weights and Threshold For "Exclusive OR" Operation



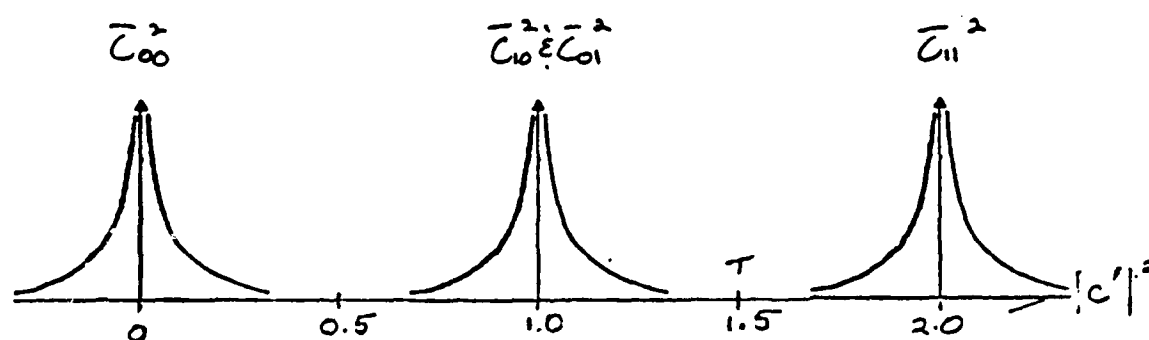
a. Equivalent Logic Gate

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

b. Truth Table



c. Complex Weights



d. Density Functions and Threshold Selection

Figure 3-13. Optimum Weights and Threshold For "AND" Logic Operation

**DYNAMIC OPTICAL INTERCONNECTIONS**

**J. W. Goodman and L. Hesselink, Principal Investigators**

**INFORMATION SYSTEMS LABORATORY  
STANFORD UNIVERSITY**

**PROGRESS REPORT - DYNAMIC OPTICAL INTERCONNECTIONS**

**Prepared For:**

**University of Dayton Research Institute**

**Contract RI-39898**

**December 20, 1985**

**Prepared by:**

**J.W. Goodman and L. Hesselink, Principal Investigators**

**Information Systems Laboratory**

**Stanford University**

**Stanford, CA 94305**

## Introduction

This report covers work accomplished on contract #RI-39898 during the time period July 1, 1985 through December 31, 1985. In fact, due to the late arrival of the contract, it was not possible to engage graduate students in the work until the beginning of Fall quarter, October 1985, and therefore the work reported took place primarily in the time period 1 October 1985 until the time of this writing, 15 December 1985. Work is now well under way, and a no-cost extension has been requested to make up for the lack of available personnel during the Summer months.

Our purpose here is to summarize the technical progress on this contract. We do so in three parts. First, a general discussion of the motivation behind the use of optics as an interconnect medium is presented. Following is a discussion of the advantages of *dynamic* or time-changeable interconnections from the point-of-view of computation and computer architecture. Thirdly, we present a description of several architectures we have conceived of during the course of this work, and one in particular that appears to have much merit for further consideration. Also included in this section will be a description of an experiment that is now under preparation in our laboratories. Lastly we present some administrative statistics pertinent to the contract.

## Why Optics for Interconnections?

A growing interest in the potential advantages of optics as an interconnect medium at various levels of computer architecture is now evident.

Already we see optics penetrating the problem of machine-to-machine interconnections, with the commercial advent of fiber-optic local area networks. The next logical step is penetration into the next lower layer of architecture, namely the problem of module-to-module communication within a single machine. In this context a module may represent a separate processor, a memory unit, or a fast peripheral device. The modern trends towards multiprocessor architectures are placing higher and higher demands on the communication capabilities within such machines.

Interest in optics for solving such communication problems stems from several sources, but most important is the relative freedom of streams of photons from interference with one another. While two streams of electrons in close proximity inevitably influence one another, generating crosstalk, no such mutual influence is exhibited by light waves. Indeed, it is even possible to pass two beams of light through one another without any measurable mutual interaction.

Other advantages for optics can also be cited, such as lower required drive power than electronic connections of comparable performance, but it can be shown that these advantages are generally dependent on the issue of mutual interference as well. For example, arbitrarily low cross talk between electronic interconnections can in principle be achieved if sufficient shielding is used, but such shielding increases the capacitance associated with the interconnection, thereby increasing the drive power required for the communication link.

A major goal of the work performed under this contract is the discovery of novel means for achieving *dynamic* optical interconnections. By "dynamic" interconnections we mean interconnections that can be rapidly changed, while still supporting high-speed communications. The realization of such techniques opens up the possibility of constructing computers having architectures that can be altered at will, with the changes taking place at rather high speeds. The vision is one of an optically reconfigurable computer, the wiring of which can be changed rapidly to meet the needs of the particular problem at hand. In the section that follows, we elaborate on the needs for reconfigurable architectures in computing.

#### **Why Reconfigurable Architectures?**

Modern trends in computer architecture emphasize increased computing power through parallel processing structures. It is recognized, however, that efficient computation on a parallel architecture requires a matching of that architecture to the structure of the problem at hand. Such a matching of the available computational resources to the structure of the problem requires a dynamically reconfigurable architecture, unless the computer is special-purpose and will always be used for problems of exactly the same structure. The need for dynamically reconfigurable interconnection networks has been emphasized in a recent review article in the computer literature (S. Yalamanchili and J.K. Aggarwal, "Reconfiguration strategies for parallel architectures", *IEEE Computer*, Vol. 18, No. 12, pp.44-61, December 1985).

Some appreciation for the need for reconfigurable interconnect networks



can be reached by considering some very specific computational problems. We present three such problems here, merely as examples of a much wider range of applications. The first two examples are taken from the work of W.D. Hillis, the chief architect of the computer known as the "connection machine" (see W.D. Hillis, *The Connection Machine*, MIT Press, Cambridge, MA, 1985), currently under development by Thinking Machines, Inc., under DARPA support. The connection machine itself is a collection of a very large number (e.g. 64,000) of small and simple processors, all of which are connected in the most simple fashion via nearest neighbors. However, a virtual interconnect network can be set up, in which messages are passed between processors in such a way as to simulate any desired architecture. However, the price paid for the fact that physical connections are only to nearest neighbors is an increased time for interprocessor communication, due to the time taken to pass messages from sources to destinations. Thus for any of the problems to be discussed in what follows, there is a great advantage in terms of computational speed if the interconnections can be established physically rather than virtually. Hence the motivation for a dynamically reconfigurable optical interconnection network.

The first example is drawn from the field of VLSI simulation. In the design of VLSI circuits, there is a strong dependence on computation as a simulation tool, to predict the performance of such circuits before they are actually fabricated. Design flaws can be detected at the simulation stage, and corrected before the expensive fabrication steps are undertaken. Such simulations are computationally intensive. Their speed can be increased by

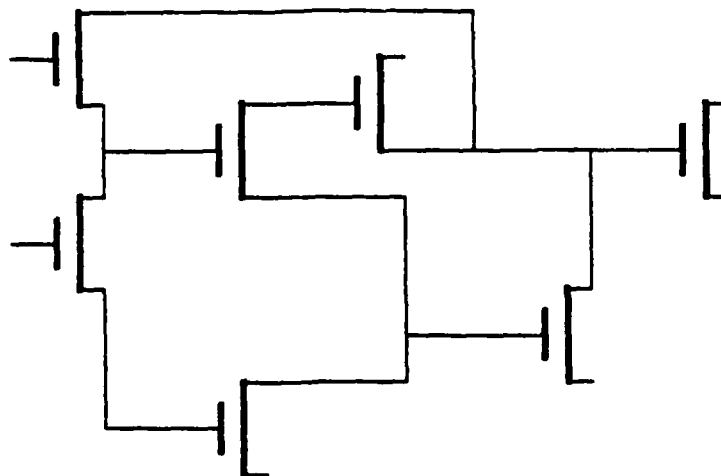
employing a multitude of processors, each working on a portion of the simulation problem. However, the interactions between these various portions of the VLSI circuit are important, and therefore it is essential that the connections between processors incorporate the constraints inherent in the actual circuit interconnections. At an extreme, a single processor can be used to simulate the performance of each transistor in the design, and the interconnections between processors can mimic the interconnections between the transistors in the circuit under simulation. Obviously if the multiprocessor computer is to be used to simulate a variety of circuits, some means must be available for changing the interconnections between processors in such a way as to reflect the various interconnections present in different integrated circuits. Thus the architecture must be reconfigurable, although the rate at which reconfiguration must take place is far smaller than the rates at which communications take place between processors. Figure 1 illustrates the close ties between the VLSI architecture being simulated and the computer architecture used for the simulations.

A second example is drawn from the field of artificial intelligence, and in particular the problem of searching semantic networks. A semantic network is a labeled and directed graph in which each vertex represents a concept, and each edge represents a relation between concepts. Thus, for example, "apple" is a concept, and both "fruit" and "computer" are concepts. Apple is connected to fruit and to computer through a relation named "is-a". Thus the term "apple" refers to either a fruit or a computer. Similarly, the concept "red" can be linked to the concept "apple" through the relation "color", but

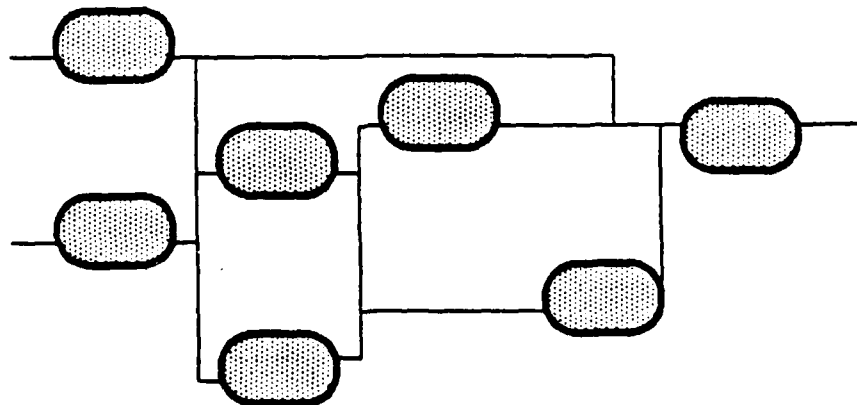
no such link need exist between "computer" and "red" (unless one has a red computer).

The problem to be attacked is the deduction of knowledge from such a semantic network. To discover specific knowledge it is necessary to search for relations between concepts of interest. One approach to this problem is to assign a single processor to each concept, and to interconnect those processors in a topology reflecting the specific semantic network under consideration. Obviously, each different semantic network to be searched requires a different interconnection topology. Hence the need for reconfigurable interconnections. Again the rates required for reconfiguration are much slower than the communication rates needed in a specific configuration. Figure 2 illustrates the parallel between the structure of a semantic network and the structure of the processor architecture applied to it.

The last example is drawn from the area of neural networks and computing (see, for example, J.J. Hopfield and D.W. Tank, "Neural computations of decisions in optimization problems", *Bio. Cybern* - to appear). There is currently much interest in the use of neural networks for solving problems of high computational complexity. Such networks, in their most common form, consist of various elements as follows: an array of neurons, which are elements that accept a sum of many analog inputs but have allowable states of only 0 or 1; a complex interconnection network between those neurons, with neuron  $i$  connected to neuron  $j$  with a weight  $T_{ij}$ . In optical versions of such networks, the basis of realization is through a parallel matrix-vector multiplier, in which the transmission of the  $ij^{th}$  matrix element represents  $T_{ij}$ , and



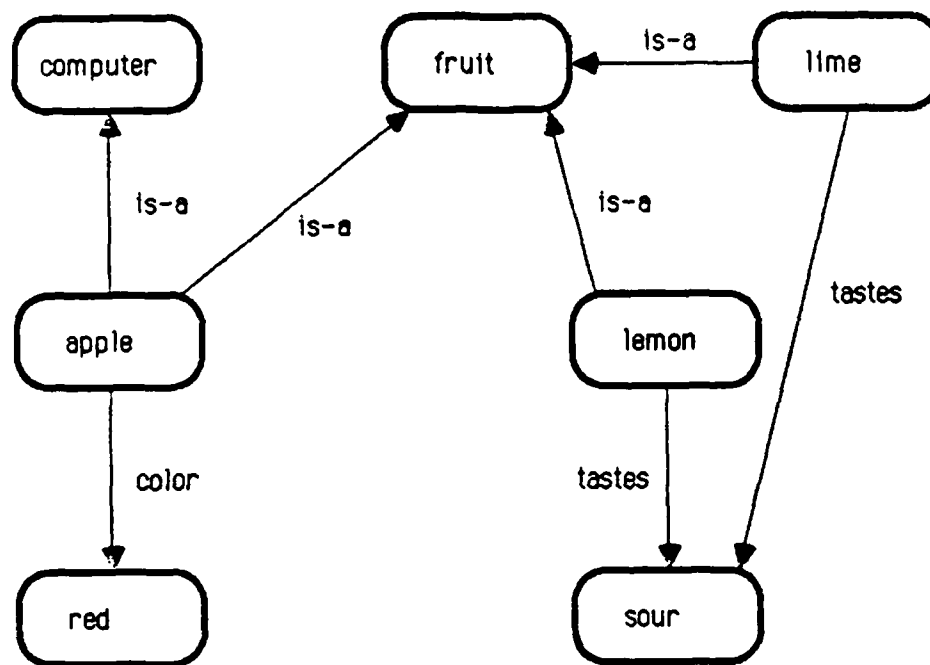
(a) The IC circuit to be simulated



(b) Processor Interconnection for simulation

*Figure 1. VLSI simulation with multiple processors*

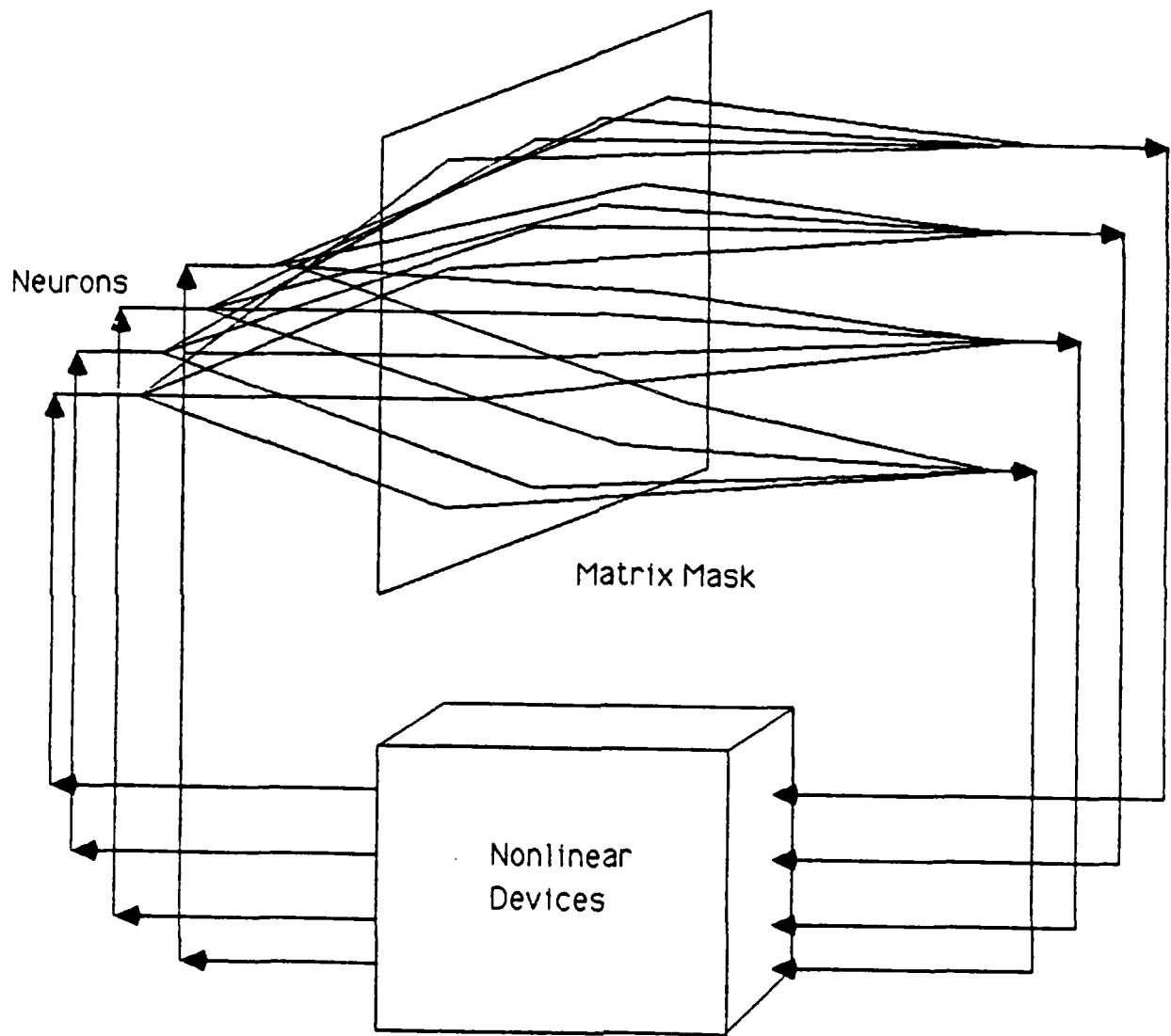
nonlinear elements are included in parallel feedback loops to force the neural states to be binary (see Figure 3 and N.H. Farhat, D. Psaltis, A. Prata and E.



*Figure 2. Searching a semantic network with a parallel computer architecture*

p.1469, 1985).

The problem to be solved by the neural network determines the interconnections required of the matrix. Thus for each new problem to be solved, a different interconnect structure is needed. Once again we see the need for dynamically reconfigurable interconnections, and once again the rate required for reconfiguration is slow compared with the rates of information transmission through the interconnections.



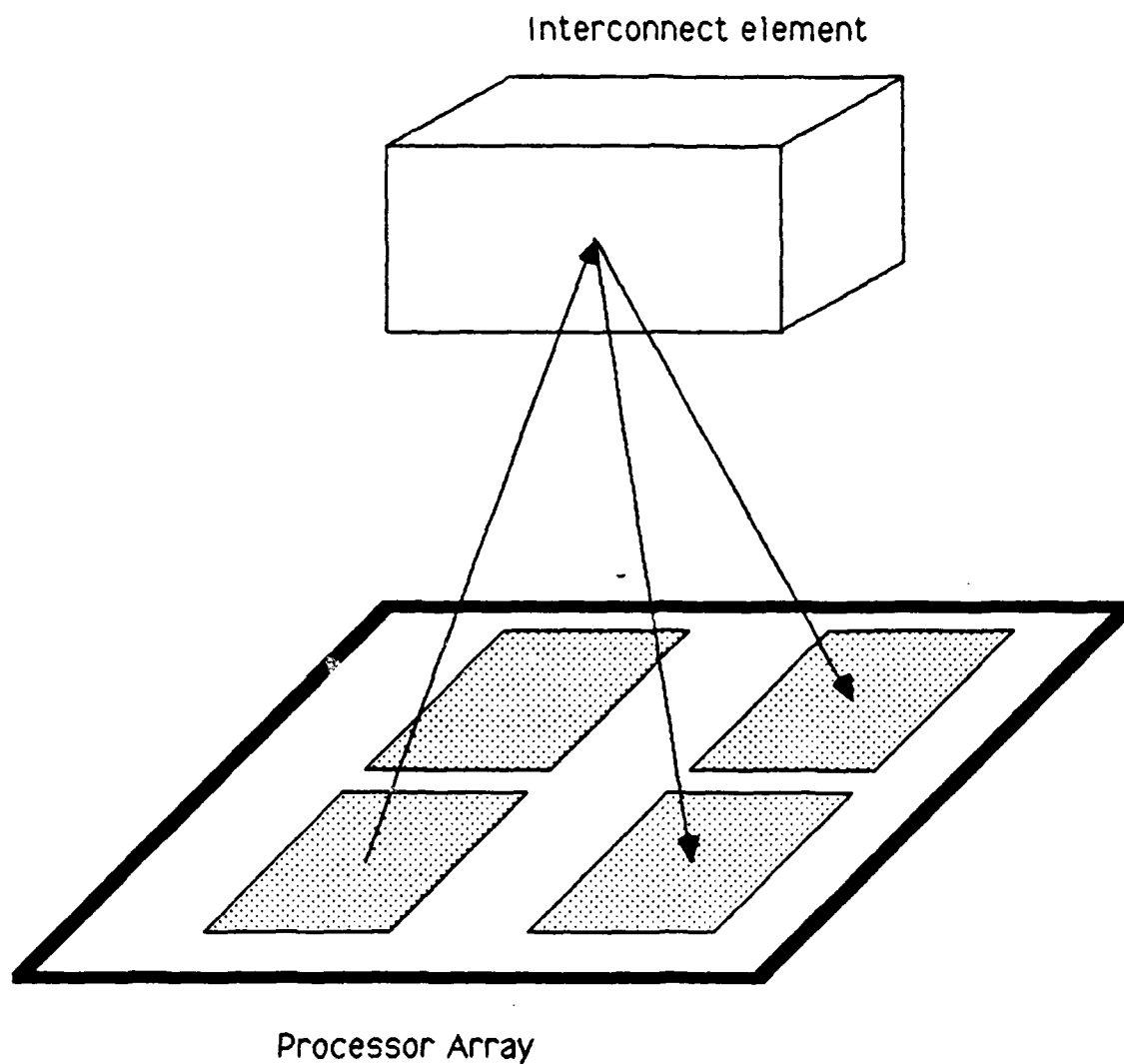
*Figure 3. Optical realization of a neural network*

With the above three examples we hope we have convinced the reader of the need for reconfigurable communication networks in modern computer architecture. The remainder of this report discusses several novel approaches to realizing such networks using optics.

### **Some Approaches to the Dynamic Optical Interconnect Problem**

In moving towards a decision as to the most promising optical approach to pursue, several ideas are worth discussion. First, in considering the ground rules to be used, two assumptions were adopted. First, the interconnect problem to be attacked should be at a high level of architecture, such as the connection of processors to processors within a single machine. Thus problems of board-to-board and chip-to-chip communication were not explicitly considered. Secondly, we assumed, for the sake of simplicity, that the processors are laid out in a plane. This is by no means an essential assumption, for indeed the solutions to be discussed are applicable to non-planar geometries. However, the planar assumption is adopted in all of the illustrations to be presented.

With the above constraints in mind, we felt that the generic form of an ideal solution might be as shown in Figure 4. A reflective interconnect element, to be specified in more detail later, resides above the processor plane. Into that reflective element is written information that establishes a pattern of reflectance, such that a source residing on one processor and illuminating the interconnect element, generates one or more reflected beams that are directed towards detectors on certain other selected processors. The optical beams



*Figure 4. Ideal interconnect configuration.*

so-directed are modulated at high rates and convey information from the processor with the active source to the processors with the active detectors. Presumably every processor contains both an optical source and a detector. It would be highly desirable if the same sources used for transmitting information could also be used for optically writing the desired reflectance pattern into the interconnect element. Thus if processor A needs to communicate



**AD-A169 882**

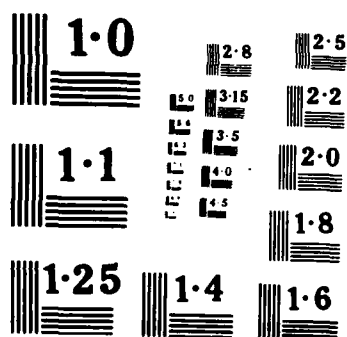
OPTICAL PROCESSING(U) DAYTON UNIV OH RESEARCH INST  
S C GUSTAFSON 31 DEC 85 UDR-TR-85-148 N00014-85-K-0479

3/6

UNCLASSIFIED

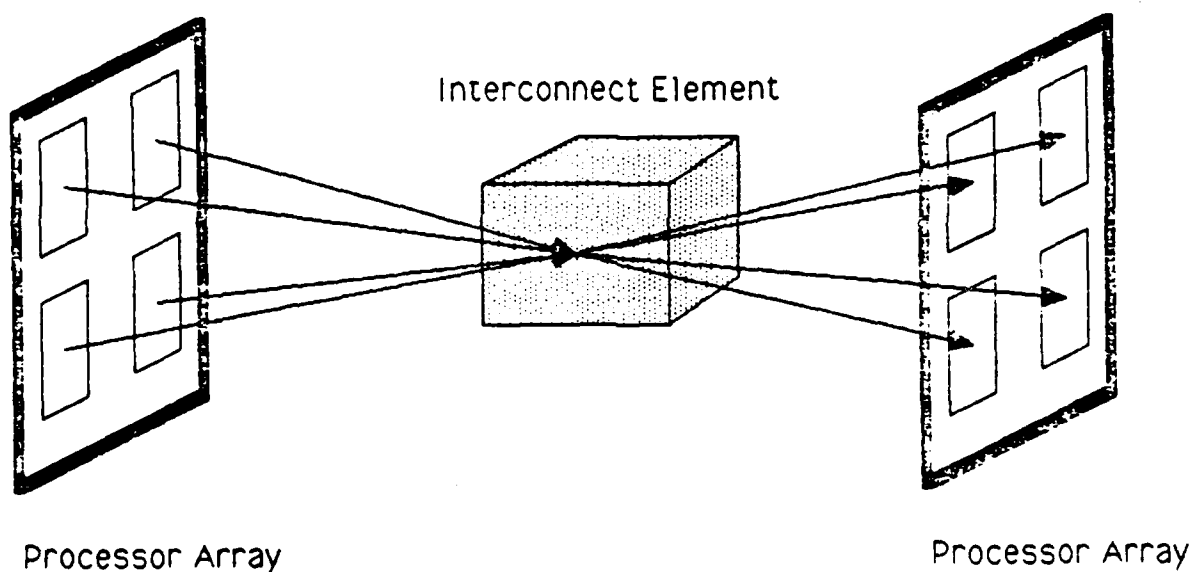
F/G 9/2

NL



with processor B, both processors illuminate the interconnect element with their respective sources. This simultaneous activation should be capable of writing an interconnect pattern suitable for establishing the desired communication link. There are many practical problems associated with such a scheme, but as an ideal, it appears to us to be worthy of consideration. Some of the practical problems will be discussed later in this report.

To move towards slightly more specific geometries, we consider the situation illustrated in Figure 5.

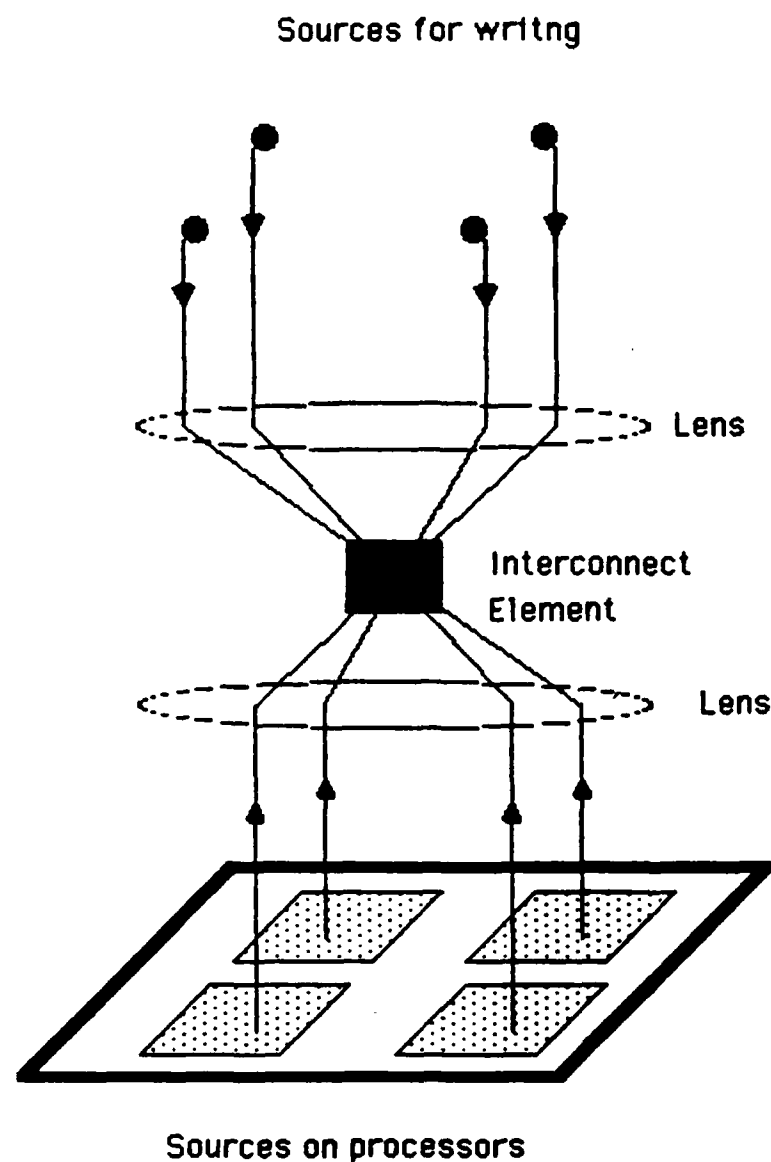


*Figure 5. Transmissive interconnect element*

In this case we have two processor arrays, one on the left and one on the right. Each processor in each array is assumed to have an associated source and a detector in close proximity to that source. The various sources are assumed to be mutually coherent, a property that would require that the

signals transmitted by the sources actually originate with a single source, with distribution to all processors, probably via optical fibers. External modulators at the processors then serve as the effective sources. To establish a communication path between processor A on the left-hand array and processor B on the right-hand array, processor A on the left illuminates the interconnect element simultaneously with processor A', also located on the left but at the position corresponding to the position of processor B on the right. The two waves from the left interfere in the interconnect element, creating a transmission hologram. Further illumination of the interconnect element with light from processor A will result in a transmitted diffracted beam that passes from the interconnect element to processor B on the right. Presumably the optical powers used in the writing phase are higher than the optical powers used in the communicating phase.

Unfortunately, the geometry discussed so far requires two processor arrays, one on the left and one on the right. We seek solutions that allow all processors to lie in a single plane. For two processors in the same plane to communicate with each other, it is necessary that a *reflective* holographic element (rather than a transmissive one) be generated. Unfortunately, in order to generate a reflective element with the right properties, a diverging spherical wave from a processor on one side of the interconnect element must interfere with a converging spherical wave coming from the opposite side of that element. A geometry for accomplishing this goal is shown in Figure 6. In this case, all the processors lie in a single plane, but sources exist on both sides of the interconnect element. The optics is assumed to generate the



*Figure 6. Geometry for writing a reflective interconnect element.*

requisite diverging and converging waves for recording the reflective focusing element. The disadvantage of this geometry is that sources on both sides of the element must be properly controlled in order to generate the needed reflective element. A much preferred solution would allow all sources to

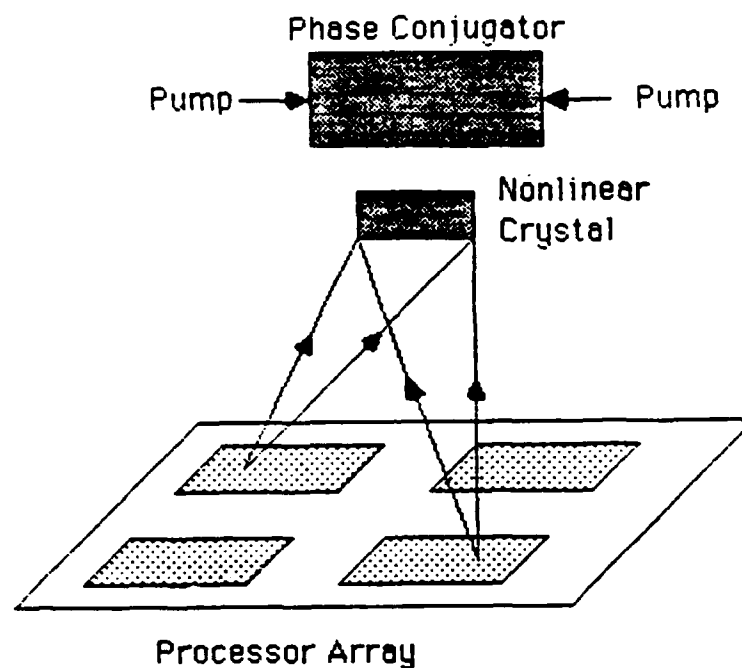
exist on one side of the element, and indeed to be the same sources used during the communication phase.

In the section that follows, we discuss in more detail a solution to the above problem. All sources can exist on the same side of the interconnect element, but the interconnect element is required to be somewhat more complex. It is this solution that we feel has greatest merit for future investigations.

#### **Dynamic Interconnect using Phase Conjugation**

The approach described in the preceding paragraph involving lenses and a procedure for writing , sequentially, mutually coherent pairs from opposite sides of the interconnect element appears to have a plane of symmetry centered around the interconnect element. In other words, it is necessary to duplicate the processor array pattern in order to be able to establish the interconnect pattern desired. Another approach therefore suggests itself, one in which this symmetry of the optical configuration is exploited by using optical phase conjugation. It would, at least in principal, be much more attractive from a layout and implementation point of view if we could use the processor array pattern itself to establish and program the interconnect pattern. A possible configuration is shown in Figure 7.

Every processor has associated with it a source and receiver and located above the processor plane is a nonlinear crystal to be used for establishing the interconnect pattern and a phase conjugate reflector. The procedure is now as follows. Suppose that processor A wishes to communicate with B, the



*Figure 7. Phase conjugation geometry*

receiver. To establish the connection B sends out a spherical wave incident on the nonlinear interconnect crystal and the phase conjugate mirror. Simultaneously two counter propagating pump beams are also incident on the phase conjugator and a phase conjugate or backwards propagating beam towards processor B results. As soon as a phase conjugate grating has been formed B is turned off. The phase conjugate mirror has memory associated with it in that a phase conjugate beam remains even after the source from B has stopped radiating. Subsequently the source from A is lit and both the converging and diverging spherical waves needed to establish the interconnect

pattern are incident on the nonlinear crystal. A grating is formed such that spherical waves coming from processor A are directed towards B. After the interconnect pattern has been established communication between the two processors may commence. The nonlinear crystal also has memory, i.e. the grating persists after the writing beams are turned off. However, during communication the grating may be gradually erased depending on the intensity of the communication waves. To establish a new pattern the previous one is erased by illuminating it with a uniform source and repeating the procedure.

The principle of operation just described needs to be investigated in detail in order to be able to answer important questions regarding the performance of this device. During the writing and establishment of the phase conjugate beam the nonlinear crystal is exposed as well. In fact when both the source from B and its phase conjugate are present a reflection hologram is set up in the nonlinear crystal. This is undesirable and can be avoided by several means. The sensitivity of the nonlinear interconnect element is much less than that of the reflector and consequently during the grating formation time in the reflector the interconnect crystal is not affected. Alternatively, we may use gain in the phase conjugate reflector during recording of the interconnect pattern and a much less sensitive interconnect crystal and thus avoid the formation of unwanted gratings. Signal amplification may be achieved by varying the intensity of the readout beam for the reflector.

The issue of grating storage in the nonlinear medium also needs to be addressed. Currently used materials such as BSO and BGO have only very



limited storage capacity; depending on the intensity of the beams used in recording and readout the memory time may be on the order of milliseconds. In ferro electrics on the other hand the grating can be fixed but the sensitivity is much smaller than for BSO and BGO. In addition these crystals are not very sensitive in the near infrared regime compatible with semiconductor sources. We are currently investigating these issues related to the nonlinear recording materials and their impact on the architecture described. It is not our intention to perform materials research, but it is necessary to study the impact of materials parameters on the performance of the device.

To address the feasibility aspect of the interconnect system as currently envisioned, we have planned several experiments. First we have prepared an experiment involving the optical architecture described with the exception that instead of nonlinear recording materials film, is used. The phase conjugate mirror is replaced by sensitive silver-halide holographic film and the interconnect element will be made of relatively insensitive film. The procedure for establishing the interconnect pattern is similar to that described, except that the recording is, of course, no longer dynamic in nature and chemical processing is needed to develop the film. The issues we are particularly interested in investigating are the effect of spurious reflections on the interconnect performance and the feasibility of the approach in terms of establishing the communication pattern. For instance in practice it may be necessary to place the sources and receivers of each processor at slightly different locations. Therefore, the resolution of the interconnect grating should be such that some degree of aberration occurs in order to receive light at the

desired location. It may in fact be possible to use the same element on the processor for both the source and the receiver, in which case this issue disappears. Currently we have designed the optical architecture for this experiment and the necessary optical components are being procured. We hope to carry out this experiment during the first quarter of 1986.

Upon completion of the preliminary experiment involving film we intend to replace the optical interconnect element with a nonlinear crystal such as BSO or BGO. In particular we are interested in investigating the effect of spurious gratings in this element due to the establishment of the grating in the phase conjugator. In particular we may be able to take advantage of the nonisotropic behavior of these crystals. For instance gratings with a grating vector aligned parallel to the direction of an applied field are more efficient than those in other directions. Therefore we may be able to selectively favor reflection gratings over transmission gratings which are formed due to spurious reflections.

#### **Administrative Matters**

The personnel involved in this investigation have been Prof. J.W. Goodman and Prof. L. Hesselink, Principal Investigators, and two graduate student research assistants. Two oral presentations were made to the Optical Computing Consortium Review Panel, one in August in San Diego, and the second in October in Washington, DC. As of yet, no other technical papers have been presented, either orally or in writing.

**THRESHOLDING AND OPTICAL COMPUTING**

Steven C. Gustafson, Principal Investigator  
Steven L. Cartwright  
David L. Flannery  
Gordon R. Little  
John S. Loomis  
L. Maugh Vail

**UNIVERSITY OF DAYTON RESEARCH INSTITUTE**

UNIVERSITY OF DAYTON  
RESEARCH INSTITUTE  
DAYTON, OHIO 45469

THRESHOLDING AND OPTICAL COMPUTING

Report on Research supported  
by SDIO/IST through  
ONR contract No. N00014-85-K-0479  
and conducted at the University  
of Dayton from 1 June to 31 December 1985

Steven C. Gustafson, Ph.D.,  
Principal Investigator  
Steven L. Cartwright, Ph.D.  
David L. Flannery, Ph.D.  
Gordon R. Little, Ph.D.  
John S. Loomis, Ph.D.  
L. Maugh Vail, Ph.D.

Abstract

Research on thresholding operations in optical computing was carried out. Specific tasks included (1) investigations of all-optical threshold elements and networks, (2) theoretical and experimental work on holographic weighting for threshold systems, (3) experimental and computer simulation studies of adaptive matched spatial filtering, and (4) initial investigations of coherence theory in optical computing. A summary of this research is followed by ten papers that describe significant experimental, analytical, and computer simulation results.

## TECHNICAL SUMMARY

### 1. OBJECTIVES

Objectives of the research at the University of Dayton Research Institute (UDRI) were to investigate thresholding operations in optical computing in the broadest and most fundamental manner consistent with the successful identification and development of certain research breakthroughs. These breakthroughs are widely believed to be necessary to realize the potential of optical computing for multiple-order-of-magnitude improvements in speed, power consumption, size, reliability, etc., compared to current and projected all-electronic computing technology.

### 2. DESCRIPTION OF WORK PERFORMED AND RESULTS

Summaries of research in the four major task areas, with the lead principal investigators indicated, are given below. More detailed descriptions are given in the ten papers that comprise the Technical Discussion.

- (a) All-optical Threshold Elements and Networks (D. L. Flannery, S. C. Gustafson, L. M. Vail).

The potential of optics-based technology for performing the basic decision and interconnection operations required in any data processing system was analyzed. Optical threshold logic designs for elementary register-level operations were developed, including 2- and 8-bit multiply-add designs and designs for signed-digit arithmetic. Circulating packet and lock-and-clock architectures suitable for current and projected bistable optical devices were also identified. Finally, electro-optically implemented Grossberg neural network models for adaptive pattern recognition were considered.

- (b) Theoretical and Experimental Work on Holographic Weighting for Threshold Systems (S. C. Gustafson, G. R. Little).

Optical processing systems characterized by thresholding operations concentrated at one functional location were analyzed. A complete design for a lumped threshold 2-bit multiplier was developed. Holographic implementations of the required weighting operations were identified for the coherent source, complex weight case. A four-input-bit experimental effort was carried out for this case that included an improved phase stabilization/control scheme. The possible use of optical phase conjugation in systems with holographically implemented weighting or interconnection operations was also assessed.

- (c) Experimental and Computer Simulation Studies of Adaptive Matched Spectral Filtering (D. L. Flannery, J. S. Loomis, L. M. Vail).

Theoretical and experimental work was carried out on a laboratory correlator that uses binary magneto-optic spatial light modulators for both the image input and a real-time programmable binary phase-only filter. An adaptive matched spatial filtering concept based on rapidly sequenced filters was defined as having high potential for real-time image processing in recognition, discrimination, and tracking tasks. Image coding using pseudorandom shift register sequences, which may be valuable in the determination of optimum correlator aperture patterns, was also assessed.

- (d) Initial Investigations of Coherence Theory in Optical Computing (S. C. Cartwright).

A consideration of real-time nonlinear optical processors led to the realization that optimum optical pro-

cessing system designs or design tradeoffs might be identified through the comprehensive application of coherence theory. The cross-spectral density function was determined to be suitable for this purpose, particularly if generalized gratings or arrays of optical bistable devices (mutually incoherent if they exhibit gain) are included in the system.

### 3. CONCLUSIONS AND RECOMMENDATIONS

Research in the areas of all-optical threshold elements and networks and phase conjugation and other nonlinear techniques for holographic thresholding and weighting will provide the best opportunities for long-range developments of fundamental importance. In contrast, it is anticipated that research on adaptive matched spatial filtering and on the application of coherence theory will provide the best opportunities for short-range developments of more immediate practical importance.

Two specific recommendations for long-range development are as follows. First, the circulating packet, lock-and-clock, and related all-optical architectures should be studied using simple "test case" problems, both experimentally and in computer simulations. Second, holographic systems for discrete pattern recognition, table look-up computation and related operations should be investigated in the same way. The incorporation of neural network or associative memory concepts and techniques such as phase conjugation in either of these areas has particular promise for achieving multiple-order-of-magnitude computer performance increases.

## TECHNICAL DISCUSSION

Attached are ten papers, two published and eight submitted or prepared for publication, that describe significant experimental, analytical, and computer simulation results. A majority of the research reported in each of these papers was supported by SDIO/IST through ONR contract No. N00014-85-K-0479 and conducted at the University of Dayton from 1 June to 31 December 1985. The papers are listed by their major task area below.

### 1. All-Optical Threshold Elements and Networks

D. L. Flannery and L. M. Vail, "All-Optical Decision Elements and Optical Computing."

S. C. Gustafson, "Thresholding and Weighting in Optical Computing," revision of book chapter submitted for publication in Optical Computing, Marcel Dekker, 1986.

S. C. Gustafson, D. L. Flannery, and R. O. Winder, "Analysis of Threshold Logic for Applications to Optical Computing," Proc. SPIE 564, pp. 173-178, San Diego, CA, Aug. 1985.

L. M. Vail and D. L. Flannery, "Robust Tracking Using Electro-Optically Implemented Neural Networks."

### 2. Theoretical and Experimental Work on Holographic Weighting for Threshold Systems

S. C. Gustafson, J. A. Kirk, G. R. Little, R. P. Kenan, and C. M. Verber, "Optical Implementations of Lumped Threshold Logic," Proc. SPIE 564, pp. 157-166, San Diego, CA, Aug. 1985.

G. R. Little, "Phase Stabilization and Control Technique with Improved Precision," to be submitted to Applied Optics.



G. R. Little, "Holographic Weighting and Phase Conjugation for External Thresholding Architectures."

3. Experimental and Computer Simulation Studies of Adaptive Matched Spatial Filtering

D. L. Flannery and J. S. Loomis, "Adaptive Matched Filtering"

S. C. Gustafson, "Image Coding Using Pseudorandom Shift Register Sequences," to be published Proc. SPIE 514, paper no. 46, presented Cannes, France, Dec. 1985.

4. Initial Investigations of Coherence Theory in Optical Computing

S. C. Cartwright, "Note on Coherence Theory and Optical Processing Systems."

ALL-OPTICAL DECISION ELEMENTS  
AND OPTICAL COMPUTING

David L. Flannery, S. C. Gustafson, and L. Maugh Vail  
University of Dayton Research Institute  
Dayton, Ohio 45469

Abstract

Some aspects of the design of optical computers using all-optical decision elements are reviewed. In particular, the circulating packet and lock-and-clock architectures are described and related to current and projected bistable optical devices. As an example, threshold logic implementations of signed-digit arithmetic are considered.

## Bistable Optical Devices and Transphasors

Bistable optical devices (BOD) and the closely related Transphasor devices are the subjects of considerable current and planned device research and development interest. The recently published book, Optical Bistability: Controlling Light with Light, by H. Gibbs (1985)<sup>1</sup> comprises a comprehensive source of both the theory and recent experimental accomplishments in this area, although it excludes the laser and laser amplifier device approaches which recently have shown promise. An attractive concept for the ultimate all-optical computer,<sup>2</sup> using optical logic elements and capitalizing on optical interconnections, is what we prefer to call the circulating packet (CP) architecture shown in Figure 1. This is only one possible concept and is itself subject to many possible variations. The comparative analysis of such candidate architectures, strongly coupled with critical assessment of proposed device concepts and their ultimate potential for CP computer implementation, is a critical need.

The CP architecture reflects the current strategy for developing an all-optical computer which will be orders-of-magnitude superior to competing electronic computers on the basis of the important figures of merit, which include not only raw thru-put (operations/second) but, probably of greater importance, specific power consumption (ops/sec/watt) and specific size (ops/sec/unit volume). It now appears that optical logic elements will at best equal electronic elements in the areas of gate density and switching speed and energy.<sup>3</sup> Thus, a reasonable strategy relies on the inherent parallelism and noninterfering propagation of optics to provide the orders-of-magnitude advantage through superior interconnections. Algorithms and architectures which capitalize on the inherent parallelism afforded by optical interconnects will be an important ingredient in attaining optical computing superiority for many problem types.<sup>4</sup>

From the preceding discussion, two required areas of device research and development are apparent: optical interconnects and

optical logic elements. The interconnect area is of obvious paramount importance since it is expected to yield the winning advantage over electronics;<sup>5</sup> it is reflected in work on various types of holographically interconnected device concepts,<sup>6</sup> such as multiplier modules.<sup>7</sup> The other area, optical logic devices, is the subject of this section. Optical input/output logic elements will be critical to the success of the optical computer; conversion between optical and electronic domains at each logic element will be unacceptable due to the amount of device overhead and the associated power dissipation involved.<sup>8</sup>

Gibbs<sup>1</sup> (p.305) has succinctly summarized the salient characteristics of an ideal discrete BOD:

- Small size - characteristic dimension of a micrometer.
- Switching energy less than 1 fJ (or holding power less than 1 mW).
- Speed - switching time of 1 ps or less.
- Room temperature operation.
- Integratable to permit large numbers of interconnections insensitive to external perturbations.

To these can be added other obvious considerations such as gain (required in the CP loop), fan-in and fan-out (cascadability), a need for an inverting logic element (NOT), and input-output isolation. Since a new computing technology is to be developed, the application of threshold logic in optical computing is worthy of investigation, based on its known advantages in terms of gate count, logic levels, and reduced interconnections.<sup>8</sup> As has been pointed out by Armstrong,<sup>3</sup> an ultimate practical issue for any logic element type, with more stringent requirements in the threshold logic case, is that of manufacturability, i.e., producing large arrays of gates with sufficiently small tolerances on threshold levels and weights. Considering the early state of development of BODs, a serious consideration of the manufacturability question would be premature. Nevertheless, such

issues are worth keeping in mind as we assess and analyze candidate logic gate approaches.

No BOD device approach has yet demonstrated the combination of characteristics stated above for an ideal computing element. However, a number of promising research results on several attractive device concepts have been reported recently (e.g., at the Third OSA Optical Meeting on Optical Bistability, Tucson, Arizona, 2-4 December 1985).<sup>3</sup> Progress is being made, and no fundamental barrier to the desired performance has been discovered. Devices having switching energies of one pJ or holding powers of several mW have been demonstrated. Area-scaled projections based on current devices indicate the potential for fJ switching energy for micrometer-dimension devices, although such scaling does not always hold. Picosecond switching times have also been demonstrated. The materials and structures under investigation have potential for integration. Recent reports<sup>3</sup> of laser and laser amplifier BOD devices offer hope that the gain requirement can be met. Thus it is reasonable to expect that, given adequate research and development effort, such devices will be available in the future.

#### Circulating Packet Architectures

BODs having picosecond switching times are required for the CP architecture, and several promising candidates are being investigated. These include GaAs etalons, with or without multiple quantum wells (MQW), GaAs Self Electro-optic Effects Devices (SEED)<sup>1</sup>, CdS etalon devices<sup>1</sup>, and InGaAsP/InP laser structures.<sup>3</sup> This list is not exhaustive but includes approaches currently showing greatest promise. Other approaches are under investigation and may show greater promise in the future.

A careful study and assessment of promising devices relative to their application in CP type architectures is advisable both to form tentative comparative rankings and (probably of more

practical value) to define critical issues to both aid assessment of current device work and to aid in planning future work. As an example of such analysis, consider the electrical connections and power dissipation which would be associated with dense arrays of two promising device types: SEED devices and Laser devices. Both types promise attractive optical and total (electrical plus optical) switching energies, if they can be scaled to micrometer sizes. Both type devices require electrical connections to each logic element, and actually dissipate at least 80% of their switching energies electrically as deposited thermal energy. However, the details are totally different for the two approaches. Using reasonable estimates of on-chip connection capability, based on electronic integrated circuit technology (e.g. conductor size, resistance, capacitance), plus projected device characteristics, the power dissipation, speed, and other pertinent parameters of device arrays can be analyzed. Such an analysis should provide interesting information and uncover issues which deserve further analysis.

A subject of legitimate concern is the performance limitation which might be imposed on a CP computer by thermal dissipation in the logic gate array plane. It is much too early to perform definitive analyses of this issue; however the following represents a reasonable projection based on currently known factors.

Assume:

- 25  $\mu\text{m}$  x 25  $\mu\text{m}$  gate element size
- 8  $\mu\text{m}^2$  gate active area
- 10 fJ/ $\mu\text{m}^2$  switching energy per unit area
- 100 ps switching cycle
- 300 ps clock period
- 50% gate duty cycle (i.e. on average 50% of gates switch each clock period)

These assumptions lead, through simple and obvious calculations, to the following performance numbers:

Gate density:	$1.6 \times 10^5$ gates/cm <sup>2</sup>
Throughput:	$2.7 \times 10^{14}$ switching events/cm <sup>2</sup> /s
Power dissipation:	21.3 W/cm <sup>2</sup>
Photons/Switching Event:	about $10^5$

Some discussion is in order. The cell size and gate area chosen are believed to be reasonably conservative. The gate area is at least ten times the diffraction limited spot area for laser diode wavelengths. The cell area is much larger than gate area, which allows room for other connections or devices which might be required with a given gate approach. This also favors isolation of the devices both thermally and electronically. The switching energy assumed is also believed to be reasonably conservative in view of recently reported results for SEED devices<sup>9</sup> and is an order of magnitude above theoretical limits claimed for that type device. The assumption of 100 ps switching time is not unreasonable as several current device approaches have reached or are approaching that speed. The 300 ps clock time is based on what seems to be a reasonable CP loop size. It corresponds to a round trip distance of about nine cm. 50% duty cycle is rather arbitrarily chosen. This factor should be strongly dependent on both the problem being solved and the algorithm used. However, duty cycles exceeding this value seem unlikely.

The resulting power dissipation density is not at all prohibitive. Values ranging from 100 W/cm<sup>2</sup> to as high as 700 W/cm<sup>2</sup> are achieved using forced liquid cooling. It must be noted that the figure given is a lower bound since it includes only the electrical and optical power dissipated in gate switching, and other sources of power dissipation may be associated with a given computer approach. The number of photons per switching event is sufficient to support threshold logic operation with 1.5% threshold tolerances with noise margins of five times the standard deviation of the photon noise.

The performance potential suggested by the above projection must be viewed as a strong incentive to pursue all-optical com-

putation technology. Throughput at least on a par with current supercomputers is projected for each square cm of gate array plane area. The volume of the CP optical package associated with that performance would be on the order of  $10 \text{ cm}^3$ . Of course any realistic assessment must include a volume and power allowance for the cooling system, input/output interface hardware, and other support requirements. If we assume that this remaining hardware requires 1000 times the volume of the core CP module, the resulting total computer volume is  $10,010 \text{ cm}^3$  or about 0.35 cubic feet. This is at least two orders of magnitude below that of a supercomputer such as the Cray.

#### Lock-and-Clock Architectures

By lock-and-clock architectures, we refer to the type of architecture proposed by S.D. Smith and co-workers as shown in Figure 2.<sup>10,11</sup> The important difference between this and the CP type architectures is the assumption that switching times, and thus minimum logic pulse durations, are longer than the round trip transit time of the optical loop. This necessitates the latching (or locking) of logic signals at certain planes, as well as precautions to prevent the continuous feeding of a signal around the loop (i.e., isolating planes). Smith<sup>10</sup> has shown that for transmission bistable latching array planes, these requirements are minimally met by using three such planes in the loop with the appropriate three phase system clock.

The type of BOD envisioned for use in the lock-and-clock architecture is the ZnSe interference filter (etalon) device. These devices are thermally switched and exhibit switching times of  $10 \mu\text{s}$  or greater. Smith and co-workers have published extensively on these devices and proposed architectures which would utilize them (e.g., Figure 2).<sup>12</sup> They have announced plans to demonstrate a lock-and-clock computer within one or two years. A clever example of another architecture to solve a specific problem with these devices has been given by Wherrett.<sup>13</sup>



## Threshold Logic Implementations of Signed-Digit Arithmetic

This section outlines a suggested implementation of signed-digit arithmetic using threshold logic in a CP all-optical computer. The modified signed-digit (MSD) arithmetic system, which uses the radix-2 subset of general signed-digit arithmetic, has already been shown to have promise for optical implementation by Drake, Bocker, Lasher, Patterson, and Miceli.<sup>14</sup> Their analysis resulted in the definition of key MSD logic modules which enable carry-free addition and subtraction in parallel pipeline configurations which capitalize on the intrinsic optical computing advantages of parallelism, non-interfering connections, and high bandwidth. In this section, the realization of the MSD key logic modules is analyzed using arrays of optical threshold logic elements in a circulating packet optical computing architecture. Preliminary results are favorable: the key modules can be realized with only a few threshold logic elements each, and proper interconnection is easily implemented in the circulating packet architecture. These results provide motivation to extend the research toward higher level functions, and to consider other architectural variations, as well as signed-digit representations of radix other than two.

The effort to date has generated threshold logic implementations of the four key modules for MSD addition and subtraction in the circulating packet (CP) architecture. The trinary digit was encoded as two binary digits, and the CP implementation involved a latency of one clock cycle to achieve the two levels of threshold logic involved. Considerable modularity was realized because a good deal of commonality was achieved between threshold logic elements used by the different key modules. Figures 3 through 6 show the threshold elements and CP architecture implementations for two key modules. The module designs are preliminary and additional effort investigating the possibility of different digit encoding schemes and alternate threshold logic element choices is justified. The goal will be to minimize both

the number and types of logic elements, the latter to achieve greater modularity.

The design of threshold logic implementations based on signed-digit arithmetic using radix other than two and the implications for hardware tolerances (e.g. threshold tolerances) of using higher radices are worthy of consideration. Based on reasonably optimized key MSD logic module designs, the design of high-level register-type devices, such as a multiply-and-add module, should be possible. This would result in useful information concerning the viability of this approach in comparison to others.

The CP architecture we have considered is only one of many possible variations; other variations may prove to be optimal for some or all applications. To review, the CP architecture we have assumed involves a single plane containing an array of threshold logic elements, and a feedback path capable of general interconnections, i.e., between any output of the logic plane and any input. We have assumed a synchronous or clocked system locked to the round-trip transit time of a data packet. No use of pipelining has been considered. Obvious variations worthy of consideration include two or more cascaded logic planes, storage of multiple data packets in the loop, and even multiple optical loop configurations. Variations on how control or inputs are introduced are obviously possible. Perhaps the number of dummy or pass-thru elements required to implement the two-level MSD key modules in the baseline CP configuration can be reduced or eliminated using the variations mentioned. This is an example of just one factor of an overall optimization problem posed by the CP type architectures. A generally important and probably very difficult task is to define the overall optimization criteria for the CP architectures, or essentially, for the all-optical computer.

### References

1. H.M. Gibbs, Optical Bistability: Controlling Light with Light, Academic, 1985.
2. S. C. Gustafson, "Thresholding and Weighting in Optical Computing," Optical Computing: Digital and Symbolic, R. Arrathoon, ed., Marcel Dekker, to be published 1986.
3. Third OSA Topical Meeting on Optical Bistability, Tucson, AZ 2-4 December 1985.
4. A. Huang, "Architectural Considerations Involved in the Design of an Optical Digital Computer," Proc. IEEE, 72, pp. 780-786, July 1984.
5. D. Psaltis, Y. Abu-Mostafa, "Theoretical Investigation of the Effective Utilization of Parallelism and Connectivity in Optical Computers," ONR/SDI Optical Processing Meeting, Washington, DC, 14 October 1985.
6. S. C. Gustafson, J. A. Kirk, G. R. Little, R. P. Kennan, C. M. Verber, "Optical Implementations of Lumped Threshold Logic," Proc. SPIE, 564, No. 22, San Diego, CA, 23 August 1985.
7. S. C. Gustafson, "Adaptive Optical Processor Algorithms, Architectures and Devices," Proc. SPIE, 551, No. 22, Arlington, VA, 9 April 1985.
8. S. C. Gustafson, D. L. Flannery, R. O. Winder, "Analysis of Threshold Logic for Application to Optical Computing," Proc. SPIE, 564, No. 24, San Diego, CA, 23 August 1985.
9. D. A. B. Miller, D. S. Chemla, T. C. Damen, T. H. Wood, C. A. Burrus, Jr., A. C. Gossard, W. Wiegmann, "The Quantum Well Self Electro-optic Effect Device: Optoelectronic Bistability and Oscillation, and Self-Linearized Modulation," IEEE J. Q. E. Vol. QE-21, No. 9, 1462, Sept. 1985.
10. S. D. Smith, A. C. Walker, F. A. P. Tooley, J. G. H. Mathew, M. R. Taghizadeh, "Demonstration of a Triple Bistable-Element Loop Circuit for a Digital Parallel All-Optical Computer," 3rd OSA Top. Meet. Opt. Bistability, Tucson, AZ, 2-4 December 1985.
11. A. C. Walker, "The Application of Bistable Optical Logic Gate Arrays to All-Optical Digital Parallel Processing," to be published, App. Opt., March 1986.
12. S. D. Smith, "Lasers, Nonlinear Optics and Optical Computers," Nature, 316, pp. 319-324, July 1985.

13. B. S. Wherrett, "All-Optical Computation: A Design for Tackling a Specific Physical Problem," Appl. Opt., 24, pp. 2876-2883, September 1985.
14. B. L. Drake, R. P. Bocker, M. E. Lasher, R. H. Patterson, W. J. Miceli, "Photonic Computing Using the Modified Signed Digit Number Representation," ONR/SDI Optical Processing Meeting, San Diego, CA 21 August 1985.

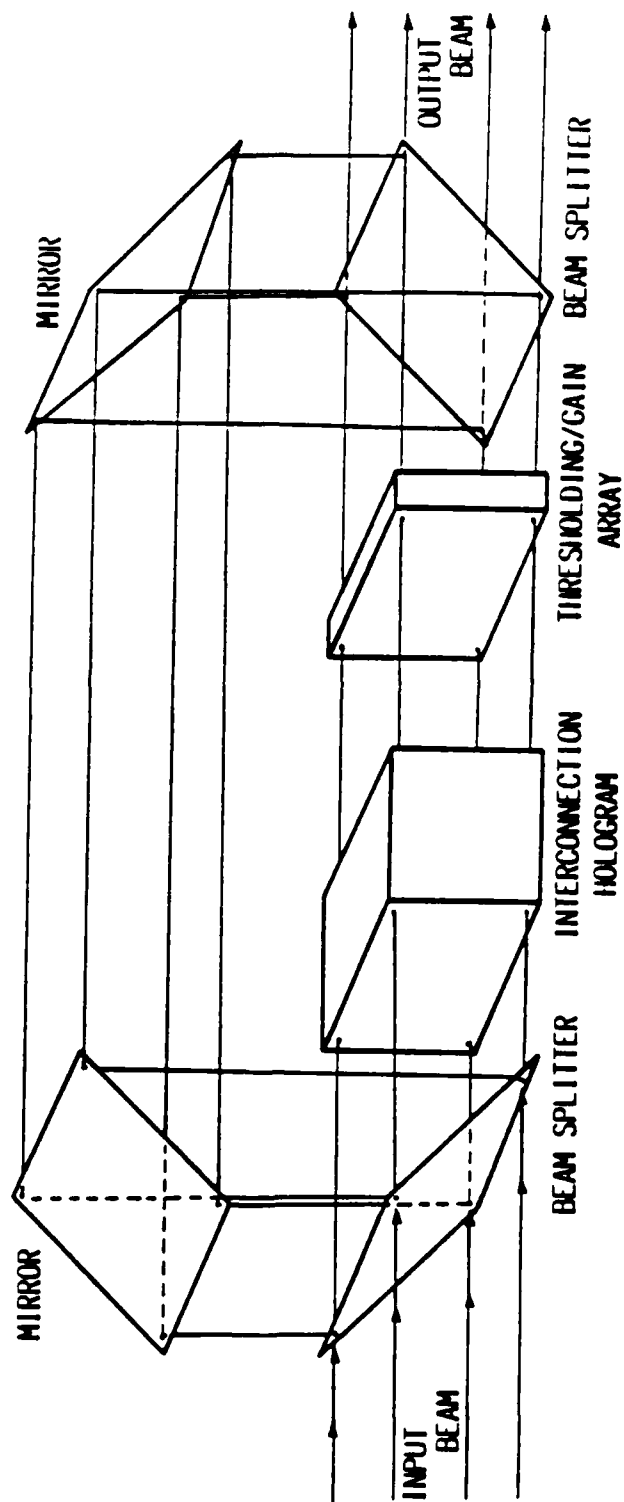


Figure 1. Circulating packet architecture for an all-optical computer.

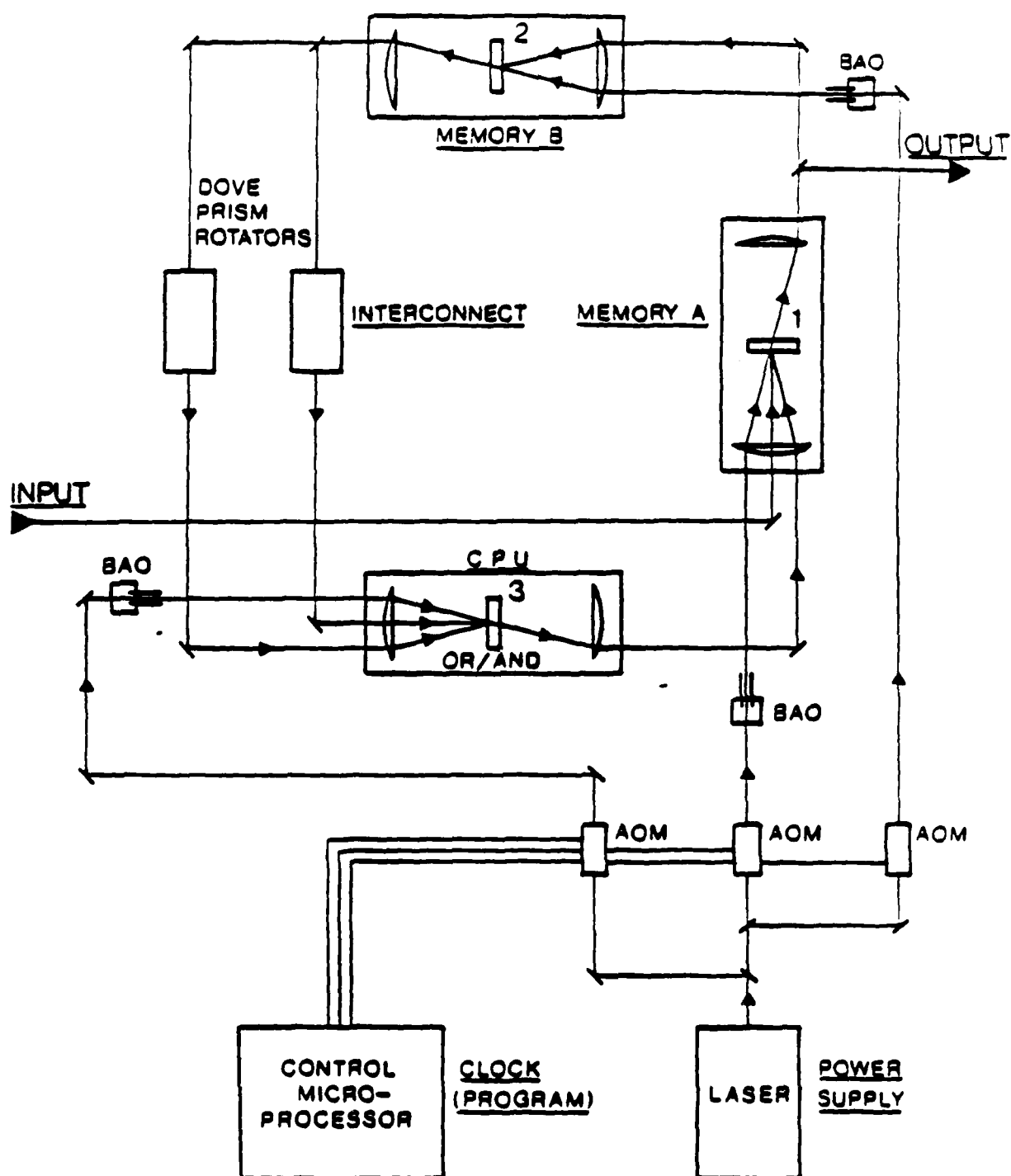
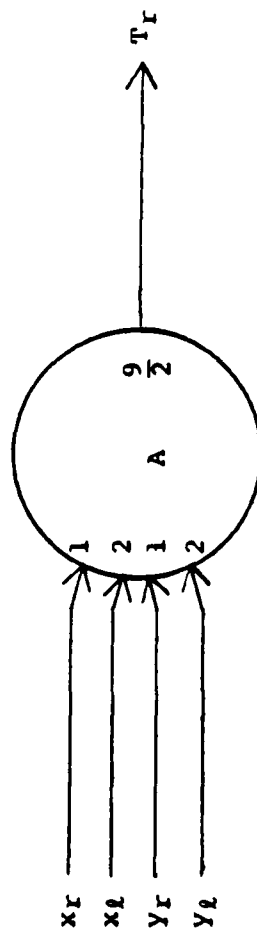
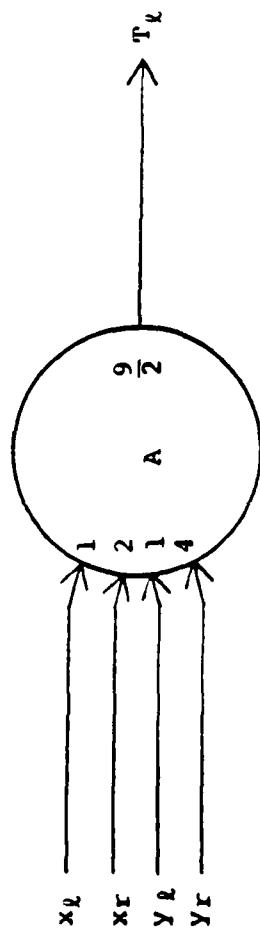


Figure 2. Lock-and-clock architecture for all-optical computer.



T  
TRUTH TABLE

X	Y			
	1	1	1	0
0	1	1	0	1
1	0	1	1	1

SD REPRESENTATION

$N_L N_R$	N
0 1	1
1 1	0
1 0	1
0 0	x

Figure 3. Threshold Logic Implementation of T-module.

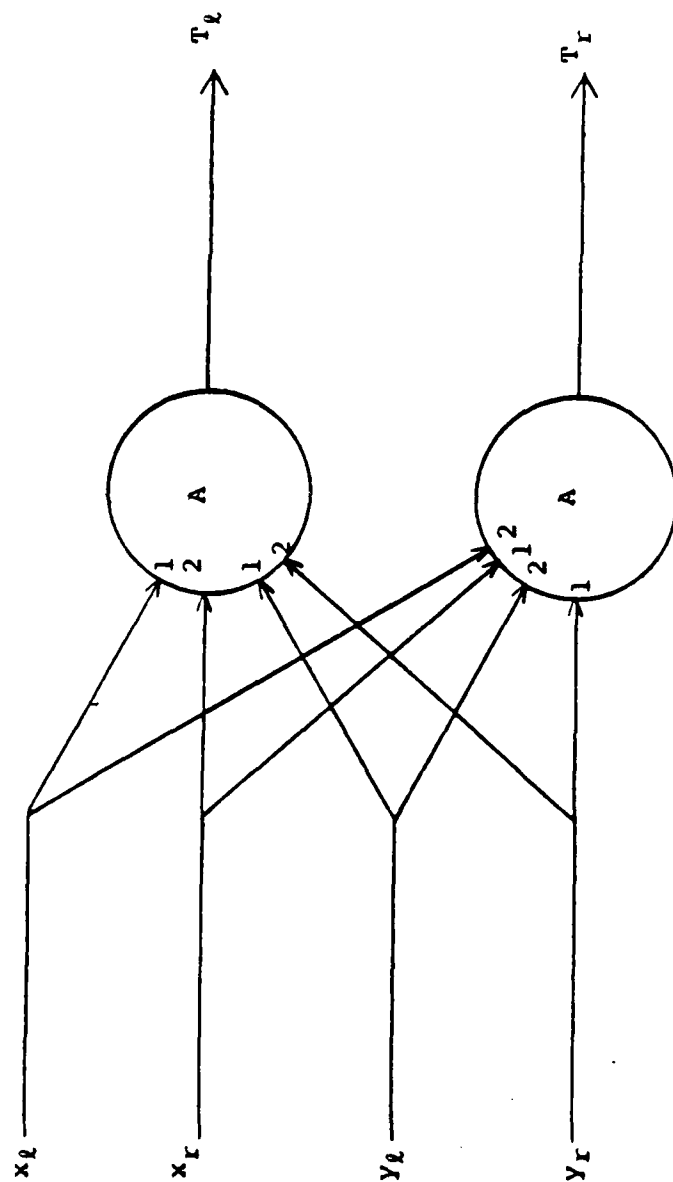


Figure 4. Circulating Packet Schematic for T-module.



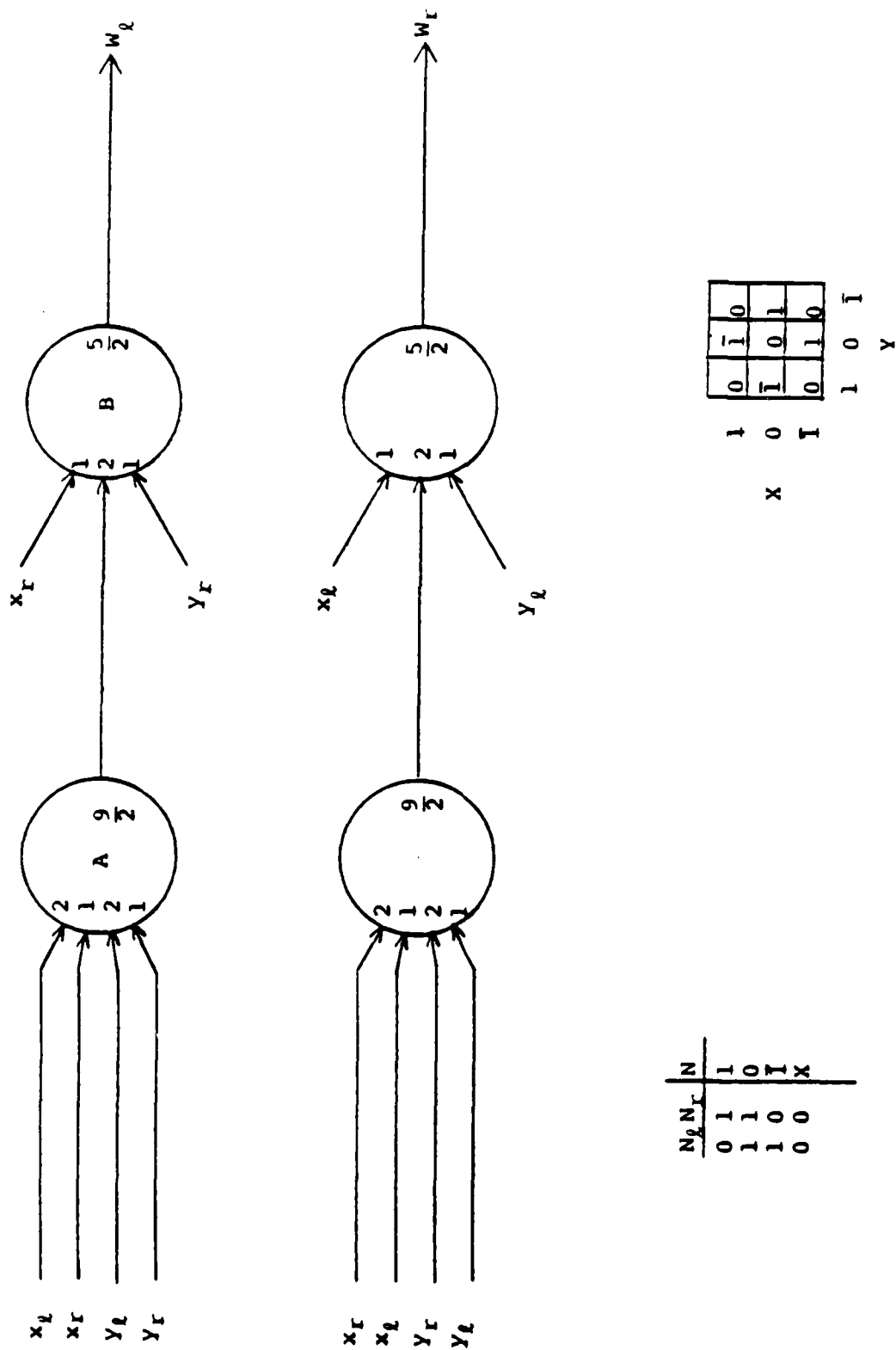


Figure 5. Threshold Logic Implementation of W-module.

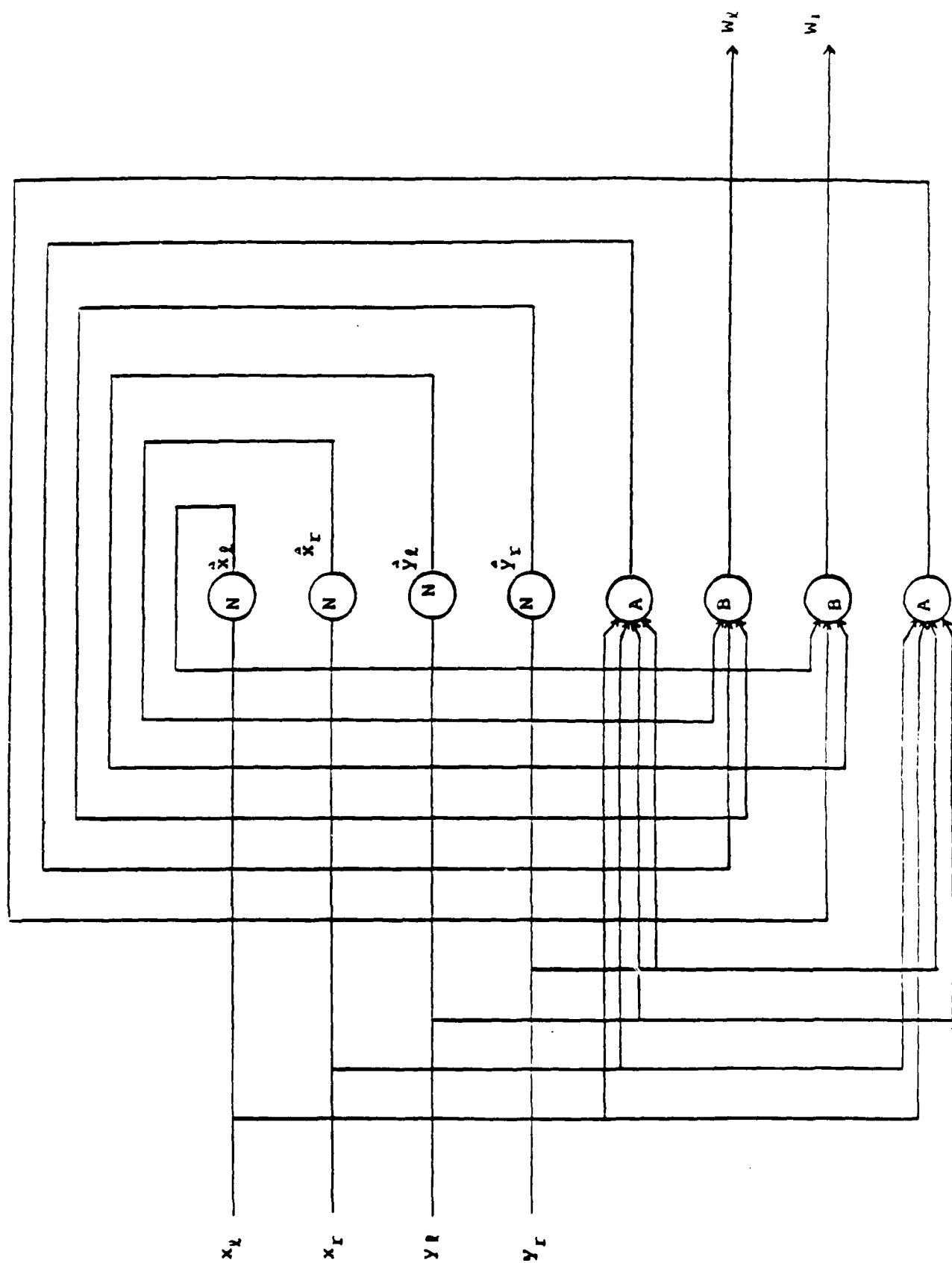


Figure 6. Circulating Packet Schematic for W-module.

## ADAPTIVE MATCHED FILTERING

David L. Flannery and John S. Loomis  
University of Dayton Research Institute  
Dayton, Ohio 45469

### Abstract

Theoretical and experimental work is reviewed and extended on a laboratory correlator that uses binary magnetooptic spatial light modulators for both image input and real-time programmable binary phase-only filtering. The corresponding adaptive matched filtering concept appears to have significant potential for real-time image processing applicable to recognition, discrimination, and tracking tasks.

## Introduction

This paper builds on the experimental and theoretical coherent correlation capability reported in "Improved Optical Filters for Image Tracking."<sup>1</sup> In that effort a laboratory correlator using binary magneto-optic spatial light modulators (SLM) for both image input and real-time programmable binary phase-only filtering (BPOF)<sup>2-5</sup> was assembled and operated. Figure 1 is a diagram of the correlator optical system. The SLM devices were Litton LIGHT-MODS<sup>TM</sup><sup>6</sup> having 48-by-48 elements. Results were in excellent agreement with a theoretical model based on the fast Fourier transform and validated the excellent correlation performance achievable with BPOF techniques. Figure 2 shows initial theoretical and matching experimental results. The work reported here was motivated by the concept of adaptive correlation illustrated by the system schematic of Figure 3. This concept applies adaptive filter selection techniques combined with a rapidly programmable Fourier plane filter (i.e. the BPOF realized with LIGHT-MOD<sup>TM</sup> devices) to yield a system capable of handling complex scenarios (which may require hundreds or even thousands of reference images) on a real-time basis. Efforts concentrated on achieving true real-time laboratory performance and defining requirements for a filter/image data bank to support planned computer simulations of the adaptive correlator concept.

## Laboratory Correlator

The results shown in Figure 2 were static in the sense that, although the spatial light modulator devices can be cycled at TV frame rates, the drive circuitry and computer software in use did not support the speed of the devices. To load and program a single image or filter pattern required about fifteen seconds, most of which involved disk access and file conversion activities. A major portion of the current effort was generation of improved software for the Apple IIe computer to speed up the input and filter programming process. As a result, the capabil-

ity is now available to sequence both filter and input images at the rate of approximately four frames per second, which is now limited by the processing capability of the Apple computer. The software, as modified, also allows at least thirty image or filter patterns to be stored simultaneously in computer memory, and allows the operator to build complex sequences of image and filter inputs with variable time delays. Another valuable feature is a shift operation which generates straight line motion of an input object. The combined effects of these new capabilities were incorporated in the production of a video tape recording demonstrating dynamic correlation using the X-O input array shown in Figure 2 for both X and O filters. Figure 4 shows computer model results for correlation using a binarized image of the NASA space telescope. These results are as expected based on previous work<sup>2-5</sup> and demonstrate the much sharper correlation peak achieved with phase-only correlation.

#### Adaptive Correlator Concept

Plans have been developed for implementing a computer model of the adaptive correlator concept shown in Figure 3. Table 1 is a set of specifications generated as goals for the image/filter memory bank, which is a vital part of such a model. The bank consists of digital images and corresponding Fourier transforms, rapidly accessible, corresponding to parametric variations of target object images. Such variations could correspond to changes in size, aspect angle, and orientation angle (i.e., scale variation and two degrees of freedom of angular variation). Ideally, views (and transforms) of the target for all possible variations of both azimuth and elevation would be required, although a discrete set of views covering every 10° in rotation and each 10 percent change in scale may be adequate. A special case is the overhead view (elevation = 90°) for which only one angular variable (rotation, or orientation angle) is required. Note that the image bank model would be used to pre-

sent sequences of filters (target object transforms) to the adaptive matched filter correlator simulation, and would also provide sequences of input scenes for the simulation. For input scenes, the target image would be superimposed on a background such as typical terrain or a mathematically simulated random pattern, with shift (translation) as a parameter. (This is necessary to simulate tracking of a moving object). In addition, the following features are required:

1. Option to provide images and filters in thresholded binary form (adjustable threshold for images; binarize on sign of real part of filters).
2. Ability to provide degraded resolution versions of images and filters (e.g., averaging over 2-by-2, 3-by-3, 4-by-4 pixel cells).
3. Ability to easily vary signal-to-noise (i.e., target-to-background) intensity ratio.
4. Ability to interface to correlation and dynamics computer models in computationally efficient manner.

The adaptive matched filtering concept (Figure 3) has significant potential for real-time image processing applicable to recognition, discrimination and tracking tasks. A computer simulation incorporating an application scenario model, correlator model, and image/filter bank, as described above, is a key research step in the assessment of this concept.

### References

1. D. L. Flannery, J. Loomis, and A. Biernacki, "Improved Optical Filtering for Image Tracking," Final Technical Report, Task A-4-1888, AF Contr. F30602-81-C-0206, RADC/OCSP, Griffiss AFB, NY (in preparation).
2. D. Psaltis, E. G. Paek, and S. S. Venkatesh, "Optical Image Correlation with a Binary Spatial Light Modulator," Opt. Eng. 23, 698 (1984).
3. J. Horner and P. Gianino, "Phase-Only Matched Filtering," Appl. Opt. 23, 812 (1984).
4. P. Gianino and J. Horner, "Additional Properties of the Phase-Only Correlation Filter," Opt. Eng. 23, 695 (1984).
5. J. Horner and P. Gianino, "Applying the Phase-Only Filter Concept to the Synthetic Discriminant Function Correlation Filter," Appl. Opt. 24, 851 (1985).
6. W. Ross, D. Psaltis, and R. Anderson, "Two-Dimensional Magneto-Optic Spatial Light Modulator for Signal Processing," SPIE 341, Real Time Signal Processing V (1982).

Table 1. Suggested specifications for an image bank model

<u>Parameter</u>	<u>Minimum Implementation</u>	<u>Ideal Implementation</u>
Image size (pixels)	128-x-128	1024-x-1024
Grey scale	6 bits	$\geq$ 8 bits
Number of different target objects	2	Many
Number of different backgrounds	2	Many
Target range (Scale) variation	5:1 10% steps	20:1 5% steps
Target angular variations	Rotation only 10° steps	Rotation, 3° steps Aspect (elevation) 5° steps



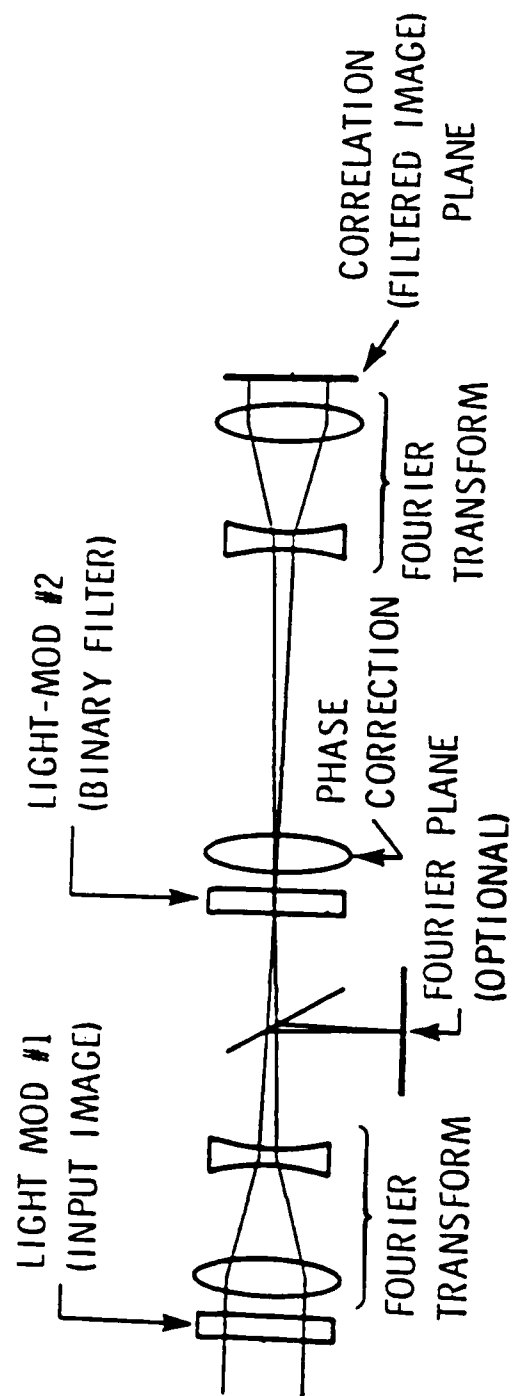
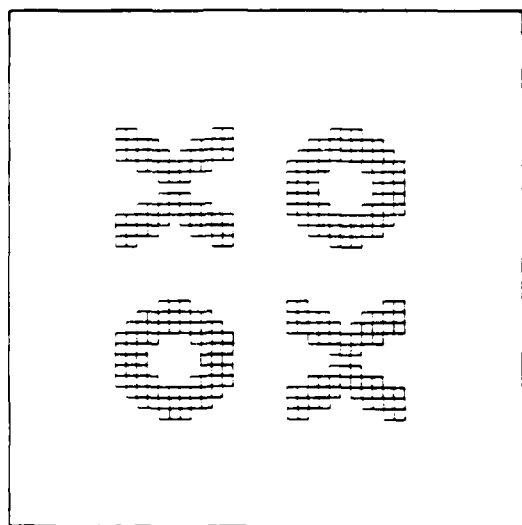


Figure 1. Optical system diagram for binary SLM optical correlator.



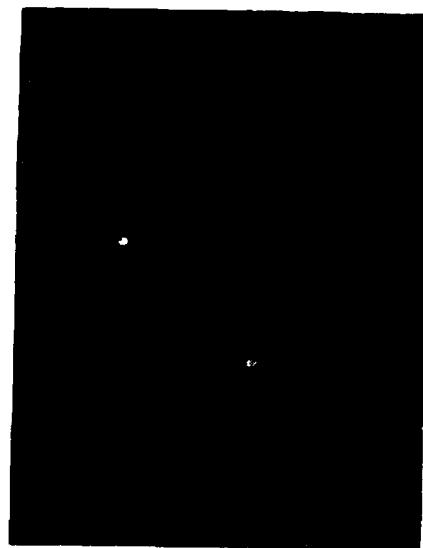
(a)



(c)



(b)



(d)

Figure 2. Theoretical and experimental results. (c) Input pattern, (b) predicted correlation with x reference pattern, (c) and (d), video monitor display of experimental results for two different brightness settings.

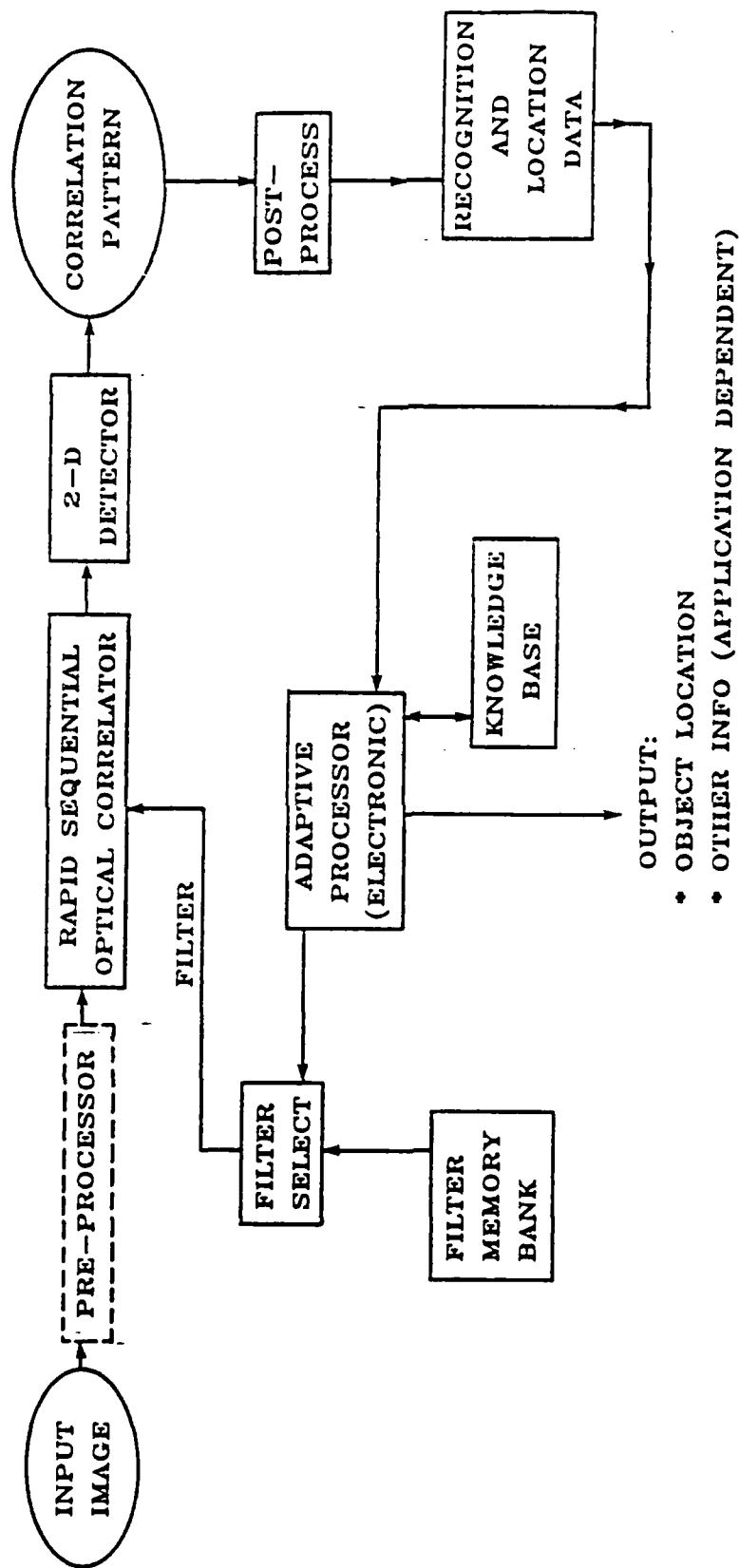


Figure 3. Schematic of adaptive correlator for target recognition, guidance, and machine vision.

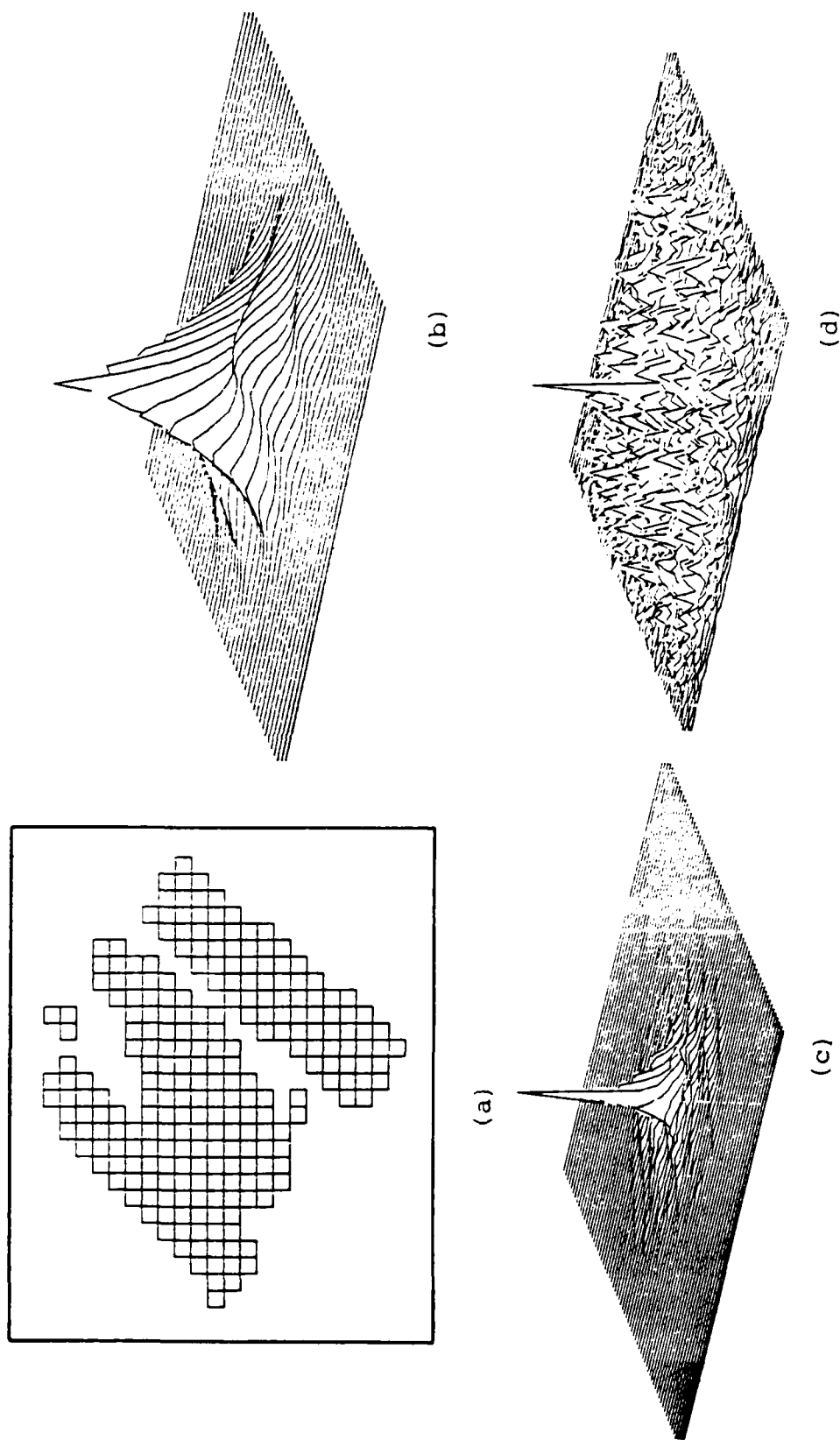


Figure 4. Computer model results for correlation using an binarized image of the NASA space telescope, (a) binary image, (b) true anticorrelation, (c) BPOF correlation, no input noise, (d) BPOF correlation, 50% input noise.

THRESHOLDING AND WEIGHTING  
IN OPTICAL COMPUTING

Steven C. Gustafson  
University of Dayton Research Institute  
Dayton, Ohio 45469

Abstract

The potential of optics-based technology for performing the basic decision and interconnection operations required in any data processing system is reviewed. Examples in which only interconnection operations are performed optically and in which both interconnection and decision operations are performed optically are discussed. A general optical computing design that summarizes key issues is also presented.

## 1. INTRODUCTION

This paper reviews the potential of optics-based technology for performing the basic decision and interconnection operations required in any data processing system. Section 1 considers motivations and definitions, Section 2 considers examples in which only interconnection operations are performed optically, Section 3 considers examples in which both interconnection and decision operations are performed optically, and Section 4 considers a general optical computing design that summarizes key issues.

Since optical computing is a relatively new concept, it is appropriate to consider architecture and programming techniques that have not thus far proved widely useful in all-electronic computers, such as residue arithmetic, multi-level logic, and threshold logic. In general, research on such nonstandard techniques has not progressed to the point where more than tentative assessments of their value in optical computing can be made. This statement also applies to the optical computing thresholding and weighting operations defined in Section 1.2. Thus this paper is necessarily brief and speculative, and all examples are simple and intended mainly to motivate future research.

### 1.1 Potential of Optical Computing

Optical computing has significant potential for multiple-order-of-magnitude performance improvements in areas such as speed, power consumption, size, memory, reliability, fault-tolerance, etc., compared to current all-electronic computing. The fundamental advantage of optics (characterized by electromagnetic frequencies of perhaps  $10^{14}$  Hz) compared to electronics (characterized by electromagnetic frequencies of perhaps  $10^7$  Hz) is the ability of optics to make relatively numerous, complex, long distance (global) and high-bandwidth intercon-

nections [1,2]. Such optical interconnections are generally (a) non-interfering; i.e., optical beams can intersect without interference but electronic "wires" cannot, and (b) relatively free of transmission line effects that can lead to significant time delays and corresponding performance constraints. Other possible advantages of optics are massive parallelism, high logic element speed, and easy three-dimensional design. These advantages are more accessible to all-electronic technology and are thus less fundamental than the interconnection advantage. In general, optics-based technology has the potential to excel in performing interconnection operations and to at least equal all-electronic technology in performing decision operations. This distinction may be attributed to the fact that the material nonlinearities required for decision operations are much more pronounced at the electromagnetic frequencies and intensities typical of all-electronic technology.

The potential of optical computing may be appreciated by considering, for example, a system consisting of a bistable optical device array in an all-optical feedback loop. Here the bistable devices perform decision operations and the feedback loop (which may contain holograms, lenses, beamsplitters, and other optical components) performs interconnection operations. As indicated in Section 4, reasonable projections of current technology may permit this system to perform on the order of  $10^{15}$  logical operations per second per square centimeter, or about 100 times higher than current VHSIC program goals [3]. A present limitation on the performance of this system is the power dissipation of the optical bistable device array; this limitation is an indication of the relative difficulty of using optics for decision operations. However, this difficulty should not preclude the development of optics-based computers with phenomenal performance compared to all-electronic computers. For example, numerous and complex

interconnections, at which optics excels, are perhaps the most important feature of biological neural network architectures in which each logic element (or neuron) may be connected to on the order of  $10^4$  other elements [4,5]. Although the individual neuron switching or decision time is long compared to typical electronic gate switching times, these biological architectures are well known to have exceptional performance characteristics, e.g., highly adaptive image recognition using small size and low-power-consumption "wetware".

## 1.2 Definition of Weighting and Thresholding

Computer architectures and operations may be represented by a set of interconnected elements that make decisions. For example, any Boolean logic function (or truth table) can be implemented by a network of basic decision or logic elements such as AND and OR gates. In general, interconnections may be characterized by weights that describe the connection strengths, e.g., 0 or 1 (for off or on) or any real or complex number. Decisions may be characterized by inequality relationships relative to a set of threshold values, which may also be real or complex numbers. Thus interconnecting may be associated with weighting and deciding may be associated with thresholding.

The functional location in space or time of the weighting and thresholding operations can be useful in classifying optical computing architectures [6]. It is appropriate to base such classification on characteristics of the decision or thresholding operations, because the most essential architec-



tural differences are generally associated with the most difficult operations. Accordingly, a first category encompasses external-thresholding architectures, where thresholding operations are performed on the non-optical side of one or more optical-electronic interfaces and where weighting operations are performed by optical elements such as holograms, lenses, etc. For example, the now-classic Stanford matrix-vector multiplier [7] (used to perform discrete linear transforms, e.g., discrete Fourier transforms) generally employs a multiple-aperture mask and passive optics to implement interconnections and an electronically thresholded photodetector array to implement decisions. A second category encompasses internal thresholding architectures, where at least some thresholding operations are all-optical and where, consequently, nonlinear devices with optical inputs and outputs are required. Thus internal-thresholding architectures contain all-optical nonlinear devices whereas external-thresholding architectures do not, and the classification separates decision operations performed optically from those performed electronically. This classification may be applied to digital or analog designs or to unclocked or clocked (e.g., combinational, asynchronous-sequential, or synchronous-sequential) architectures at levels of sophistication ranging from single gates to complete processors (e.g., gate, register, and processor levels) [8]. Note that architectures in both classes generally make essential use (through lenses, holograms, etc.) of the non-interfering interconnection capability of optics.

### 1.3 Standard and Non-Standard Threshold Logic

Threshold logic is a subject closely related to thresholding and weighting operations in computing. It was an active area of research in the 1960's and early 1970's [9-11], but since then and until very recently [12] attracted little

attention, largely because conventional Boolean logic became standard in all-electronic integrated circuit design.

Standard or linear inequality thresholded logic is illustrated in Figure 1. Here the logic function  $y = x_1x_2x_3x_4 + x_1x_3$  (for which a truth table is given) is implemented by a threshold logic element, where  $x_1, x_2, x_3$ , and  $x_4$  are binary inputs,  $y$  is a binary output, the plus sign is an OR operation, the implied products are AND operations, and the bar is a NOT operation. The element multiplies each binary input by a real-valued weight ( $w_1, w_2, w_3$ , or  $w_4$ ), sums the results, and compares with a real-valued threshold  $T$ . If the sum is less than the threshold the output is a zero; otherwise it is a one. Note from the example that a single threshold element implements a logic function that would require several conventional Boolean gates and two levels of logic (three if the NOT operation is included). Note also that the threshold element will operate properly if the threshold is any value satisfying  $4 < T \leq 5$  and that non-zero weight tolerances may be specified if the threshold tolerance is further restricted.

In general, linear-inequality thresholded logic elements have several binary inputs, one binary output, and an analog internal mechanism that may be appropriate for optical implementations (some element designs allow discrete internal mechanisms [13] and multiple binary outputs [14]). As Figure 1 illustrates, proper operation of threshold elements can often be made insensitive to fluctuations in nominal weight and threshold values which may occur due to environmental conditions, fabrication variations, etc. Note from the truth table in Figure 1b that appropriate weight and threshold values may be obtained by solving a set of simultaneous linear inequalities. For example, the twelfth row of the truth table requires

$$w_1x_1 + w_3x_3 + w_4x_4 \geq T \quad (1)$$

Since there are generally more inequalities than unknowns (in the example, 16 inequalities or truth table rows in the 5 unknowns  $w_1, w_2, w_3, w_4$ , and  $T$ ) solutions will not always exist. However, solutions that exist and that maximize various threshold or weight tolerances can be obtained using linear programming techniques. Much early work in threshold logic focused on finding such solutions or their characteristics without explicitly using linear programming, which can be computationally intensive. Advances in computer hardware and algorithms, however, have made the use of linear programming techniques [15] for the design of threshold logic elements and networks much more attractive.

As indicated above, not all logic functions are realizable using single linear-inequality threshold logic elements; those that are realizable are called threshold or linearly separable functions. In general, there are  $2^{2^n}$  logic functions of  $n$  binary variables (because each of  $2^n$  input truth table rows can have either binary output), but the number of threshold functions is usually much smaller--an upper bound is  $(2^{n^2+1})/n!$ . For example, if  $n = 3$  the total number of functions is 256, the above upper bound is 170, and the actual number of threshold functions is 104 [10]. For  $n=2$  (the simplest case) it may be easily shown that 14 of the 16 possible two-input Boolean logic gates, including AND and OR, are realizable using single threshold elements, and thus linear-inequality threshold logic may be viewed as a generalization of ordinary Boolean logic. Since any combinational (constant truth table) logic function may be realized using a network of gates or elements with no more than two levels of Boolean logic (i.e., with input-output paths passing through no more than two logic gates in series, not considering the NOT operation), it follows that the same is true for threshold logic. However, Boolean logic networks for complex functions (e.g., 16-bit multiplication) generally require more than two

logic levels to avoid interconnecting impractically large numbers of logic elements on the same level [16]. Threshold logic and, in particular, optical implementations of threshold logic may mitigate such requirements. This characterization and the example of Figure 1 indicate that threshold logic has the potential advantages of fewer logic levels (which implies greater speed), fewer logic elements (which implies less power consumption), and fewer interconnections (which implies lower complexity).

These potential advantages will usually be more pronounced for the more general nonstandard or nonlinear-inequality threshold logics. An element in such logic compares a threshold value with a nonlinear function of the binary inputs, such as a quadratic polynomial where the coefficients are considered to be weights [17]. As discussed in Section 2, coherent optical systems can implement, for example, a nonlinear function proportional to the squared magnitude of the sum of the binary inputs multiplied by complex weights, where the weights represent optical wave amplitudes and phases. In this case it has been shown, using an exhaustive but limited-resolution numerical search, that the number of logic functions of  $n = 3$  binary variables using one threshold element with complex weights and either inverted or non-inverted output is at least 246 [18]. This number, when compared with the 104 such functions that can be implemented using standard threshold logic, indicates the increased "logic power" of one nonstandard threshold logic. Some neural network architectures for optical computing [4,5,19,20] may be described in terms of such general threshold logics or may have enhanced performance if they are employed. For example, if the number of patterns of  $n$  binary pixels to be classified into one of two categories is less than  $2^n$ , then well-known adaptive methods based on the Widrow-Hoff algorithm or on perceptron architectures can usually identify a linearly

separable function or single standard threshold element that performs the classification [10]. However, if single non-linear threshold elements (of the sort that may have effective optical implementations) are allowed, then the proper classification of significantly more than  $2n$  patterns may be possible [21].

## 2. EXTERNAL-THRESHOLDING DESIGNS

In external-thresholding designs optical techniques are used only to implement interconnection or weighting operations; decision or thresholding operations are performed electronically (e.g., by photodetectors with thresholded amplifiers). The weighting operations are generally performed by some combination of diffracting, refracting, and reflecting elements in bulk or integrated optical implementations. Generalized diffraction gratings or holograms are basic components in models that use content addressable memory concepts and holographic table look-up techniques to generate desired weighting patterns [22]. Sections 2.1 and 2.2 consider two simple examples of these models: a design for a two-bit multiplier and a design that realizes any logic function of two variables.

### 2.1 Two-Bit Multiplier Example

A truth table for the multiplication of two two-bit numbers  $x_1x_0$  and  $y_1y_0$  to obtain  $z_3z_2z_1z_0$  is given in Figure 2. Suppose that the four input bits are represented by 0 if they are zero and by  $x_1 = A_1 \exp(i\phi_1)$ ,  $x_0 = A_2 \exp(i\phi_2)$ ,  $y_1 = A_3 \exp(i\phi_3)$ , and  $y_0 = A_4 \exp(i\phi_4)$  if they are ones. If these expressions correspond to waves in an approximation where all source-source, source-detector, and detector-detector distances are large compared to the wavelength, then the two-bit multiplier may be designed as shown in Figure 3a [6]. Here  $x_0$ ,  $x_1$ ,  $y_0$ , and  $y_1$  are optical point sources, and  $z_0$ ,  $z_1$ ,  $z_2$ , and  $z_3$  are point

photodetectors. The lines indicate optical paths, each of which may have a selected attenuation and phase shift that could be implemented by a hologram or by integrated optical diffracting elements.

The required attenuations and phase shifts may be obtained by solving sets of simultaneous nonlinear inequalities derived from the truth table. For example, the 12th row and the  $z_2$  column of the table (boxed in Figure 2a) imply a signal at photodetector  $z_2$  that must equal or exceed a threshold  $T_2$ :

$$|A_1 \exp(i\phi_1) + A_3 \exp(i\phi_3) + A_4 \exp(i\phi_4)|^2 \geq T_2 \quad (2)$$

Similar expressions may be obtained so that each of the four output columns in the table (labeled  $z_3$ ,  $z_2$ ,  $z_1$ , and  $z_0$ ) is described by a set of 16 (one for each table row) simultaneous nonlinear inequalities in 9 unknowns: four amplitudes ( $A_1$ ,  $A_2$ ,  $A_3$ , and  $A_4$ ), four phases ( $\phi_1$ ,  $\phi_2$ ,  $\phi_3$ , and  $\phi_4$ ), and one threshold ( $T_1$ ,  $T_2$ ,  $T_3$ , or  $T_4$ ). Solutions are required for each of these four overdetermined inequality sets (which involve terms such as  $A_1^2$ ,  $2A_1A_2\cos(\phi_1-\phi_2)$ , etc.) such that the amplitudes, phases, and thresholds obtained all have acceptable tolerances or ranges over which they may vary without affecting proper two-bit multiplier operation.

One solution which involves only phase shifts (no attenuations) and which may have practical tolerances is given in Figure 3b [6], where the  $\phi$  column gives the phase shifts required for each of the four paths (in order) to each detector, the  $T$  column gives the threshold value for each detector when  $A_1 = A_2 = A_3 = A_4 = 1$ , and the  $\Delta T/R$  column gives the fraction of the total signal range on each detector over which its threshold may vary. Figure 4 is a histogram of  $\Delta T/R$  for output  $z_2$  generated by selecting each of the four phases for this output randomly from normal distributions with means at

their design values and standard deviations equal to  $\arctan(.1)$ . (.1). These standard deviations correspond to 10% displacements of the base vectors, and Figure 4 shows that such variations reduce the threshold tolerance  $\Delta T/R$  for output  $z_2$  from 37% to about 20%. Similar acceptance tolerances may be obtained for the other outputs.

The two-bit multiplier design described above is based on the ability of optics to provide noninterfering interconnections which (a) are parallel in that interconnection time is essentially independent of interconnection length or weight and (b) lead to system operation times essentially limited only by the response times of sources or detectors. These interconnections may be provided by passive diffracting elements in the form of one or more ordinary or bulk thin or thick holograms in which light, coherent, or possibly non-coherent (white) [23], propagates approximately normal to the hologram plane. These interconnections may also be provided in integrated optical implementations by passive diffracting elements formed on or near a substrate surface such that light propagates approximately parallel to the surface. Such integrated optical implementations could use surface relief or photorefractive mechanisms to form the diffracting elements on GaAs,  $\text{LiNbO}_3$ , glass or other substrates. These implementations, have potential for realizing significant size, power consumption, and reliability advantages and for providing real-time programmable interconnections or weightings using electronically modulated diffracting element structures [6,24].

Optically generated holograms can implement the weightings required for certain truth tables or input-output relationships (including the two-bit multiplier) in external-thresholding systems. Using standard models of the holographic process [25] it may be shown [6] that an  $L \times P$  output truth table matrix  $A$  is related to an  $N \times P$  input truth table matrix  $C$  by

$$A = OR^+C, \quad (3)$$

where  $O$  and  $R$  are  $L \times M$  and  $N \times M$  matrices describing the complex amplitudes used in recording an  $M$ -fold exposed hologram and  $+$  is the conjugate transpose operation. An important aspect of Equation (3) is that although many exposures may be used to record the hologram, the ability of the hologram to represent input-output relationships is described by no more than the  $NL$  complex elements of  $OR^+$ . In the two-bit multiplier, for example, where  $N = L = 4$  and  $P = 16$ , only 16 complex parameters are available to relate 64 input bits to 64 output bits. This suggests that not all possible truth tables are realizable in an optically recorded hologram. An analogous situation (discussed in Section 1.3) is that not all logic functions can be implemented by single threshold logic elements.

It would be useful to solve Equation (3) at least approximately for  $OR^+$  in terms of  $C$  and  $A$ . This matrix equation is generally over-determined, and least-squares or pseudoinverse methods might be used to obtain an approximate solution. The (row by row) least-squares solution, for example, is

$$OR^+ = AC^+(CC^+)^{-1}. \quad (4)$$

While this solution may not yield the desired truth table realization in an external-thresholding system, it may serve as a starting point for a steepest descent or other computer search for desired solutions. Such solutions should maintain the desired input-output relationship when the matrix elements are varied over acceptable tolerance ranges. The optics phase-only solution for the external thresholding 2-bit multiplier described in Figure 3b is such a solution and may be used to derive an  $OR^+$  matrix in which all elements have unit magnitude [6]:



$$OR^+ = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & (-1 + \sqrt{3}i)/2 & 1 & (-1 - \sqrt{3}i)/2 \\ (-\sqrt{15} + i)/4 & (\sqrt{15} + i)/4 & (\sqrt{15} + i)/4 & (-\sqrt{15} + i)/4 \\ 1 & 1 & -1 & 1 \end{bmatrix} \quad (5)$$

A particular implementation of this solution for holographic recording is  $O =$  the  $4 \times 4$  identity matrix and  $R = (OR^+)^+$ . Note that although the above analysis implies three-dimensional holographic systems, integrated optical assemblies of diffracting elements similar in function to bulk holograms might be used. This possibility is related to the observation that the multiple truth table "images" to be recorded and reconstructed, although often highly cross-correlated, may be relatively simple or low-resolution bright-spot dark-spot patterns.

## 2.2 Two-Binary-Variable Example

Optical or computer generated holographic synthesis of weighting or interconnecting operations will generally require knowledge of the diffracting patterns on the hologram that modify amplitudes and phases to yield the correct truth table input-output behavior with maximum weight and threshold tolerances. In the case of the geometrical optics two-bit multiplier design of Figure 3, expressions governing input-output behavior were easily obtained. This favorable situation may be uncommon in the design of the generally smaller, more efficient, etc., external thresholding systems for which geometrical optics approximations do not apply.

Consider, for example, the derivation of eight far-field holograms, each with the same two design parameters, that implement, in an external thresholding system, the eight positive-threshold two-Boolean variable functions (i.e., the eight out of the sixteen functions for which two zero inputs yield a zero output). Figure 5 shows a simple format [6] con-

sisting of a screen with two pinholes separated by a distance  $y$ . One pinhole is covered by a phase-shifting film  $\theta$ ; there is a detector  $d$  and lower and upper mutually coherent point sources  $l$  and  $u$ . In the far-field approximation the distances  $b$  and  $y$  and the wavelength  $\lambda = 2\pi/k$  must be small compared to the distance  $s$ . With this approximation and with  $b$  fixed, the problem reduces to finding values of  $y$  and  $\theta$  such that the detected signals  $I_l$  for only source  $l$  on,  $I_u$  for only source  $u$  on, and  $I_b$  for both sources on have all six possible inequality relationships. Referring to Figure 5 for definitions, the following approximate expressions may be derived:

$$A_l = \exp i[k(w + s)] + \exp i[k(v + r) + \theta]$$

$$A_u = \exp i[k(w + s)] + \exp i[k(u + r) + \theta]$$

$$I_l = |A_l|^2 \approx (ks)^2(x^2 + ax)^2 + 2\eta(ks)(x^2 + ax) + \eta^2 \quad (6)$$

$$I_u = |A_u|^2 \approx (ks)^2(x^2 - ax)^2 + 2\eta(ks)(x^2 - ax) + \eta^2$$

$$I_b = |A_l + A_u|^2 \approx 2(ks)^2 [(x^2 + ax)^2 + (x^2 - ax)^2] \\ + 8\eta(ks)x^2 + 4\eta^2 - 4(ks)^2(ax)^2$$

where  $x = y/s$ ,  $a = b/s$ , and  $\eta = \phi - \pi \approx 0$ . Figure 6 is a graph [6] of the approximate expressions for  $I_l$ ,  $I_u$ , and  $I_b$  versus  $x$  for  $\lambda = 628$  nm,  $b = 10$   $\mu$ m,  $s = 10$  cm, and  $\eta = .004$ . Note that four of the six inequality relationships can be satisfied using the plotted values; the other two relationships can be satisfied for other values of  $\eta$ .

The example of Equation (6) indicates the possible complexity of general (physical optics) external thresholding synthesis. Greater complexity may be anticipated if Fresnel rather than Fraunhofer diffraction conditions are allowed and if the input-output truth tables are large. One approach to

such synthesis problems is to perform additional post-photodetection processing and to employ logical reduction and residue arithmetic techniques to reduce the amount of information that must be stored for holographic look-up [22]. A more general approach is to seek alternatives to requirements for conversion into and out of residue arithmetic and for additional all-electronic processing.

This approach could involve (a) obtaining the generally large sets of overdetermined simultaneous nonlinear inequalities that fully describe a desired external thresholding system, (b) finding optimal solutions, perhaps with respect to weight and threshold tolerances, for these sets using nonlinear programming techniques [26], and (c) identifying optical systems, perhaps based on computer generated holograms, that implement the solutions. In the case of optically recorded holograms, recent work indicates that control of the relative phases of the truth table look-up reference beams is important for obtaining selected reconstructions without interference (from nonselected reconstructions) [27]. Other recent work has shown that such interference can be reduced by using techniques involving gain competition in nonlinear optical resonators which contain, for example, phase conjugating mirrors [28]. These techniques clearly involve optically implemented decisions and are thus in the internal thresholding category.

### 3. INTERNAL THRESHOLDING DESIGNS

Internal thresholding designs have optical implementations for decision as well as interconnection operations. They are therefore more general than external thresholding designs, for which optically implemented decision operations are not permitted. As was indicated in Section 1, optics-based systems that implement decision operations need not make use of the same Boolean logic gate networks that have become conventional in all-electronic integrated circuits. In par-

ticular, threshold logic constitutes a more general approach that includes conventional Boolean logic as a special case and that requires no more and usually significantly fewer logic levels, elements, and interconnections to carry out the same function.

Two simple examples of internal thresholding designs, one (a multiplier-adder) involving combinatorial logic and one (a J-K flip-flop) involving sequential logic are considered in Sections 3.1 and 3.2. These designs may have effective integrated (or near-integrated) optical implementations involving nonlinear optical (e.g., bistable) devices. Since optical-electronic (or electronic-optical) conversions are generally costly in terms of speed, power consumption, device size, etc., it is anticipated that these implementations will require all-optical (or nearly all-optical) internal connections to realize substantial performance improvements compared to all-electronic designs.

### 3.1 Multiplier-Adder Example

This example was motivated by the need for high-speed and otherwise superior inner-product-step operations in linear algebra (e.g., for matrix-vector or matrix-matrix multiplication). The inner product step involves multiplying two numbers and adding a third number. A two-bit multiplier-adder, for example, multiplies two two-bit input numbers  $M$  and  $N$ , adds the result to a five-bit input number  $X$ , and outputs the results as a five-bit number  $Y$ . In clocked operation, the output  $Y$  could be fed back to the input  $X$  to achieve a multiply-accumulate result with the capability of accumulating up to three products without overflow.

A conventional Boolean logic design for a two bit multiplier-adder is diagrammed in Figure 7, where the subscripts on  $M$ ,  $N$ ,  $X$ , and  $Y$  designate binary number position

(20, 21, etc.). Note that two well-known and frequently occurring multigate configurations have been grouped and represented by single symbols. The exclusive OR (XOR) function may be carried out in two logic levels using two AND gates, one OR gate, and two inverters. The full-adder function may be carried out in two logic levels with a minimum of five AND gates, two OR gates, and four inverters. A threshold logic design for a two-bit multiply-adder [29,30] is diagrammed in Figure 8 and is composed entirely of threshold logic elements, with fan-in and fan-out limited to five. Weights are indicated inside each element symbol adjacent to the input lines, and the threshold is indicated adjacent to the output line.

Note that the Boolean logic design in Figure 7 requires a total of 38 logic gates and 18 inverters. (Inverters are not normally included in logic element or level counts; however, they do require space, time, and power). This design has a maximum propagation path of 9 logic levels. The threshold logic design in Figure 8 requires 18 threshold logic elements and involves only 5 logic levels. It is also of interest to compare the number of interconnect lines required by the designs. This count is 116 for the Boolean logic design versus 70 for the threshold logic design. The comparison may thus be summarized by stating that the threshold logic design is superior by factors of roughly two with regard to number of logic levels, number of logic elements, and number of interconnections.

A similar design comparison for perhaps a more useful case considers an 8-bit multiplier-adder that multiplies two 8-bit numbers, adds a 21-bit number, and outputs a 21-bit result. In this case, the fan-in and fan-out constraints were increased to eight. The results of this design (with the two-bit multiplier-adder results in parenthesis) are summarized in Figure 9 [29,30]. Note that the threshold logic advantage in

gate count ratio has increased to almost three-to-one, while the logic level and interconnection ratios have remained at about two-to-one. This design is reasonably complex, and one is tempted to conclude that the results are indicative of what may be obtained for more general complex designs. If viewed in terms of the ratio of processing speed to power consumption, it can be argued that the results also indicate an advantage considerably greater than just the separate level, element, or interconnection ratios, since these ratios each contribute to either increased processing speed or reduced power consumption or both.

As indicated above, it is anticipated that all-optical elements and internal connections will be required for internal thresholding systems that realize their potential for commanding and enduring performance advantages compared to all-electronic designs. If size advantages are also to be realized, integrated or near-integrated [31] optical technology constitutes one viable approach. Among the many material systems that have been investigated in this technology ( $\text{LiNbO}_3$ , glass/ $\text{SiO}_2$ / $\text{Si}$ , etc.), GaAs/GaAlAs systems have the best potential for the complete integration of optical sources, thresholding devices, and detectors on one substrate. A problem is that the material nonlinearities required for thresholding operations using all-optical devices are two to four orders of magnitude above the values that characterize current uniform electro-optic materials for the case where the optical input is provided by laser diode sources [32,33]. However, GaAs/GaAlAs multiple quantum well (MQW) structures may have the required nonlinearity properties at room temperature, with favorable switching speed and energy characteristics, and in dense arrays on GaAs integrated optical structures [34]. Other promising material systems are also being investigated, including systems based on InSb [35,36] and on multiple-layer Langmuir-Blodgett organic films [37].

### 3.2 J-K Flip-Flop Example

The J-K Flip-flop is a basic unit in conventional sequential logic architectures that is clocked synchronously by an external source or asynchronously by internal component time delays. It has two inputs J and K and an output Q: if both J and K are 0 the output is the previous value of Q, if only J is 0 Q is 0, if only K is 0 Q is 1, and if both J and K are 1 Q is the opposite of its previous value. Figure 10 shows how a J-K flip-flop could be implemented using a system in which "femtosecond pancakes" of light store previous output values in an optical feedback loop [18]. Note that a maximum of six optical paths and three all-optical thresholding devices are required and that corresponding weight and threshold values for J-K flip-flop operation are indicated.

The "femtosecond pancake" architecture [20,38,39] is equivalent to a network of threshold logic elements as in Figure 8 but with feedback. Since this architecture is among the most general and powerful designs, its characteristics and potential are discussed more generally below.

### 4. SUMMARY

The "femtosecond pancake" architecture, for which Figure 11 gives a general outline, is a suitable design for summarizing key issues related to thresholding and weighting in optical computing. In this design input data, including control and programming information, is spatially and temporally encoded on an input beam. An optical interconnection array (containing holograms, lenses, etc.), performs weighting operations, and an array of nonlinear optical devices (generally with gain) performs thresholding operations. A feedback optical path makes the overall design a sequential computing system in which spatially and temporally modulated "femtosecond pancakes" of light may circulate in a pipeline

manner. Timing is performed asynchronously with a clock period related to the optical feedback loop time or synchronously using external clock inputs. Electrical inputs to the nonlinear device array and to the interconnection array (perhaps through electro-optically controlled gratings for the latter) may also supplement, at relatively low data rates, the optical input, control, and programming data.

As noted in Section 3.2, the "femtosecond pancake" architecture is equivalent to a network of threshold logic elements with feedback. It is also equivalent in some respects to designs in which all-electronic logic gates are interconnected using optical techniques; e.g., the intra-chip optical interconnection of VLSI circuits [40]. Finally, this architecture has recently been investigated in connection with neural networks [41,42] which, as mentioned in Section 1.1, can have exceptionally "intelligent" and flexible performance characteristics. For any of these interpretations (and generally for architectures that use arrays of nonlinear optical devices) power dissipation is a current performance-limiting factor. For example, a maximum power dissipation of  $10 \text{ W/cm}^2$ , a minimum device area of  $10 \mu\text{m}^2$ , and a minimum device switching energy per unit area of  $1 \text{ fJ}/\mu\text{m}^2$ , which are reasonable projections of current GaAs-based technology [32], imply that on the order of  $10^{15}$  logical operations per second per square centimeter ( $10^{15} \text{ gate-Hz/cm}^2$ ) could be performed, which is two orders of magnitude higher than present VHSIC program goals. However, if power dissipation is of no concern, as may be the case for low duty cycle or "burst" operation, then minimum device switching times of 10ps imply on the order of  $10^{18} \text{ gate-Hz/cm}^2$ .

This chapter has discussed the potential of optical computing in terms of weighting and thresholding operations and has presented simple examples of external-thresholding and internal-thresholding designs. A major point is that optics-



based technology excels in performing interconnection or weighting operations but may have no general advantage over all-electronic technology in performing decision or thresholding operations. A research goal is thus to identify optical computing architectures that make maximum use of the interconnection and related advantages of optics to realize commanding and enduring performance characteristics.

## REFERENCES

The references below are selected (with some exceptions) to be among the most recent or comprehensive of several possible sources; they are not intended to be complete or to indicate priority.

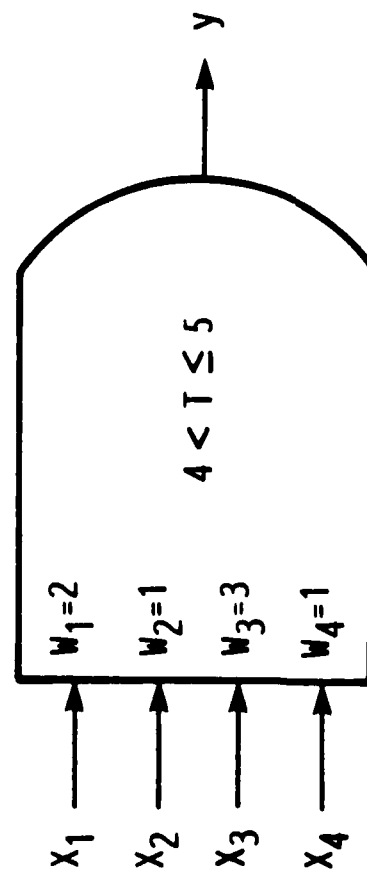
1. Optical Computing, Special Issue, Proc. IEEE, v. 72, No. 7, Jul 1984.
2. Optical Computing, Special Issue, Opt. Eng. v. 23, No. 1, Jan-Feb 1985.
3. K. Julian, "Defense program pushes microchip frontiers", High Technology v. 5, pp. 41-57, 1985.
4. H. Szu, "Neural network models for computing", ONR Optical Processing Meeting, San Diego, CA, 21 Aug 1985.
5. R. A. Athale, "Optical Attentive Associative Memory", Opt. Soc. Am. Meeting, Washington, DC, 14-18 Oct 1985.
6. S. C. Gustafson, J. A. Kirk, G. R. Little, R. P. Kenan, and C. M. Verber, "Optical implementations of lumped threshold logic", Proc. SPIE v. 564, No. 22, 1985.
7. J. W. Goodman, A. R. Dias, and L. M. Woody, "Fully parallel, high-speed incoherent optical method for performing discrete Fourier transforms", Opt. Lett. v. 2, pp. 1-3, 1978.
8. S. C. Gustafson, "Adaptive optical processor algorithms, architectures, and devices", Proc. SPIE, v. 551, No. 22, 1985.
9. M. L. Dertouzos, Threshold Logic: A Synthesis Approach, MIT Press, Cambridge, MA, 1965.
10. P. M. Lewis II and C. L. Coates, Threshold Logic, Wiley, NY, 1967.
11. S. Muroga, Threshold Logic and Its Applications, Wiley, NY, 1971.
12. R. Arrathoon and M. H. Hassoun, "Optical threshold elements for digital computation", Opt. Lett., v. 10, pp. 143-145, 1984.
13. A. Pal, "A four-variable programmable universal logic module using digital summation threshold logic gates", Proc. IEEE, v. 72, pp. 1813-1815, 1984.

14. L. A. M. Bennett, "Realization of logic functions by multi-output threshold-logic gates", IEE Proc., v. 129E, pp. 239-242, 1982.
15. K. G. Murty, Linear Programming, Wiley, NY, 1983.
16. S. Muroga, Logic Design and Switching Theory, Wiley, NY, 1979.
17. R. Arrathoon, "Digital-optical computing - possibilities and pitfalls", Proc. SPIE, v. 564, No. 17, 1985.
18. S. C. Gustafson, "Optical processing research", presented at Innovation Science and Technology Consortium on Optical Computing, Washington, DC, 14 Oct 1985.
19. D. Psaltis and N. Farhat, "Optical information processing based on an associative-memory model of neural nets with thresholding and feedback", Opt. Lett., v. 10, pp. 78-100, 1985.
20. D. Psaltis, "Theoretical investigation of the effective utilization of parallelism and connectivity in optical computers", ONR Optical Processing Meeting, San Diego, CA, 21 Aug 1985.
21. T. Kohonon, Self-Organization and Associative Memory, Springer, Berlin, 1984.
22. T. K. Gaylord, M. M. Mirsalehi, and C. C. Guest, "Optical digital truth table look-up processing", Opt. Eng., v. 24, pp. 48-58, 1985.
23. F. T. S. Yu, White-Light Optical Signal Processing, Wiley, NY, 1985.
24. R. Arrathoon, E. R. Schroeder, and L. D. Hutcheson, "Integrated electro-optic Bragg processors for digital real-time signal processing", Opt. Lett. v. 10, pp. 244-246, 1985.
25. R. J. Collier, C. B. Burckhardt, and L. H. Lin, Optical Holography, Academic, NY, 1971.
26. G. P. McCormick, Nonlinear Programming: Theory, Algorithms, and Applications, Wiley, NY, 1983.
27. C. C. Guest and T. K. Gaylord, "Phase stabilization system for holographic optical data processing", Appl. Opt. v. 24, pp. 2140-2144, 1985.
28. G. J. Dunning, E. Marom, Y. Owechko, and B. H. Soffer, "All optical associative memory with feedback using phase

conjugate mirrors", Opt. Soc. Am. Meeting, Washington, DC, 14-18 Oct 1985.

29. R. O. Winder, Private communication, Feb 1985.
30. S. C. Gustafson, D. L. Flannery, and R. O. Winder, "Analysis of threshold logic for application to optical computing", Proc. SPIE, v. 564, No. 24, 1985.
31. S. C. Gustafson, "Electro-optical intensity modulation of near-surface light for optical signal processing applications", Proc. SPIE, v. 517, pp. 163-166, 1984.
32. N. Peyghambarian and H. M. Gibbs, "Optical bistability for optical signal processing and computing", Opt. Eng. v. 24, pp. 68-73, 1985.
33. C. Verber, Private communication, Mar 1985.
34. D. S. Chemla, D.A.B. Miller, P. W. Smith, "Nonlinear optical properties of GaAs/GaAlAs multiple quantum well material: phenomena and applications", Opt. Eng. v. 24, pp. 551-564, 1985.
35. S. D. Smith, I. Janossy, H. A. MacKenzie, J. G. H. Mathew, J. J. E. Reid, M. R. Taghizadeh, F.A.P. Tooley, A. C. Walker, "Nonlinear optical circuit elements as logic gates for optical computers: the first digital optical circuits", Opt. Eng. v. 24, pp. 569-574, 1985.
36. S. D. Smith, "Lasers, Nonlinear optics and optical computers", Nature v. 316, pp. 319-324, 1985.
37. J. A. Neff, "Future directions in optical processing", NASA Conf. 2302, Hampton, VA, Aug 1983.
38. A. Huang, "Digital techniques in optical computing", Opt. Soc. Am. Meeting on Optical Computing, Incline Village, NV, 18-20 Mar 1985.
39. W. T. Rhodes, "Optical computing research architectures and algorithms", ONR optical processing meeting, San Diego, CA, 21 Aug 1985.
40. R. Kostuk, J. W. Goodman, and L. Hesselink, "Optical imaging applied to microelectronic chip-to-chip interconnections", Appl. Opt., v. 24, pp. 2851-2858, 1985.
41. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", Proc. Nat. Acad. Sci., v. 19, pp. 2554-2558, 1982.

42. Y. S. Abu-mostafa, "Efficiency of computation in neural networks", Opt. Soc. Am. Meeting, Washington, DC, 14-18 Oct 1985.



$$w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 \begin{cases} < T, y = 0 \\ \geq T, y = 1 \end{cases}$$

(a)

$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

(b)

Figure 1. Example of a threshold logic element that implements the function  $y = \bar{x}_1x_2x_3x_4 + x_1x_3$ . (a) Threshold element, (b) truth table.

(a)

<u>x<sub>1</sub></u>	<u>x<sub>0</sub></u>	<u>y<sub>1</sub></u>	<u>y<sub>0</sub></u>	<u>z<sub>3</sub></u>	<u>z<sub>2</sub></u>	<u>z<sub>1</sub></u>	<u>z<sub>0</sub></u>
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

(b)

$$\left| A_1 e^{i\theta_1} + A_3 e^{i\theta_3} + A_4 e^{i\theta_4} \right|^2 \geq T$$

Figure 2. (a) Two-bit multiplier truth table, (b) nonlinear inequality for boxed entries in truth table.

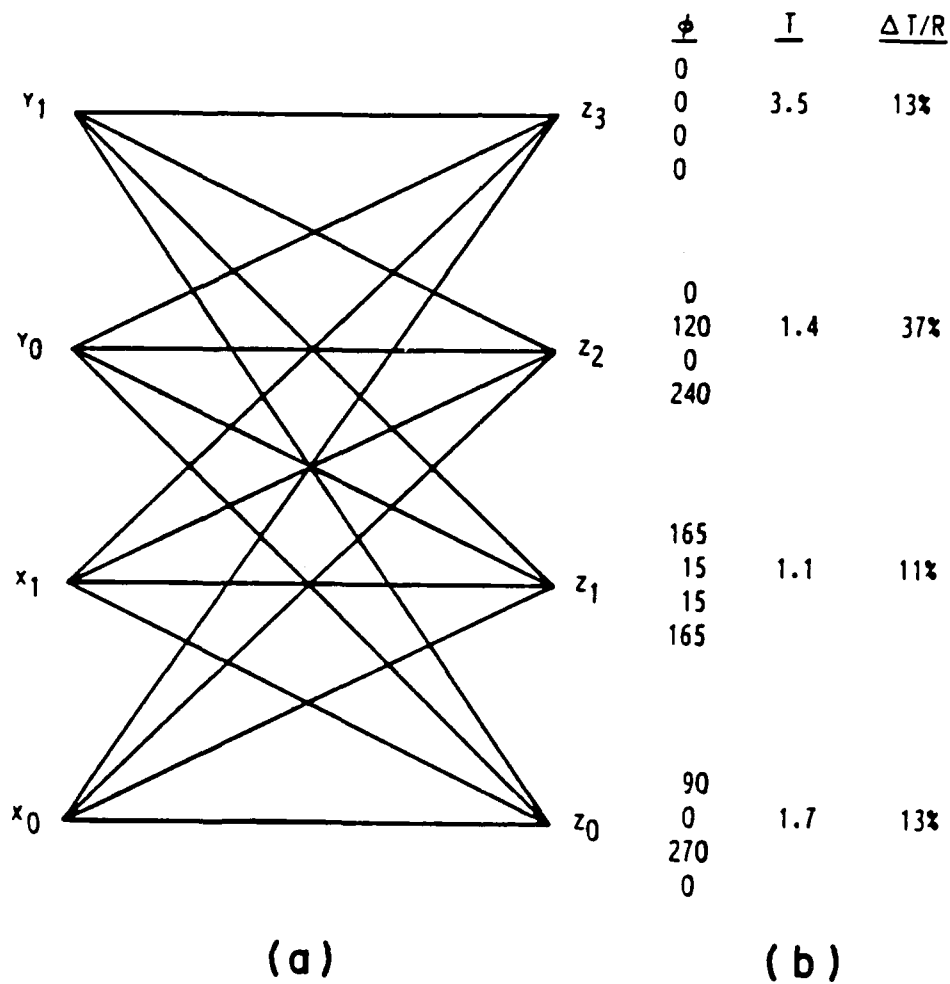


Figure 3. External thresholding two-bit multiplier. (a) Interconnections from sources  $x$  and  $y$  to detectors  $z$ ; (b) no-attenuation solution for interconnection phases  $\phi$ , detection thresholds  $T$ , and threshold tolerances  $\Delta T/R$ .



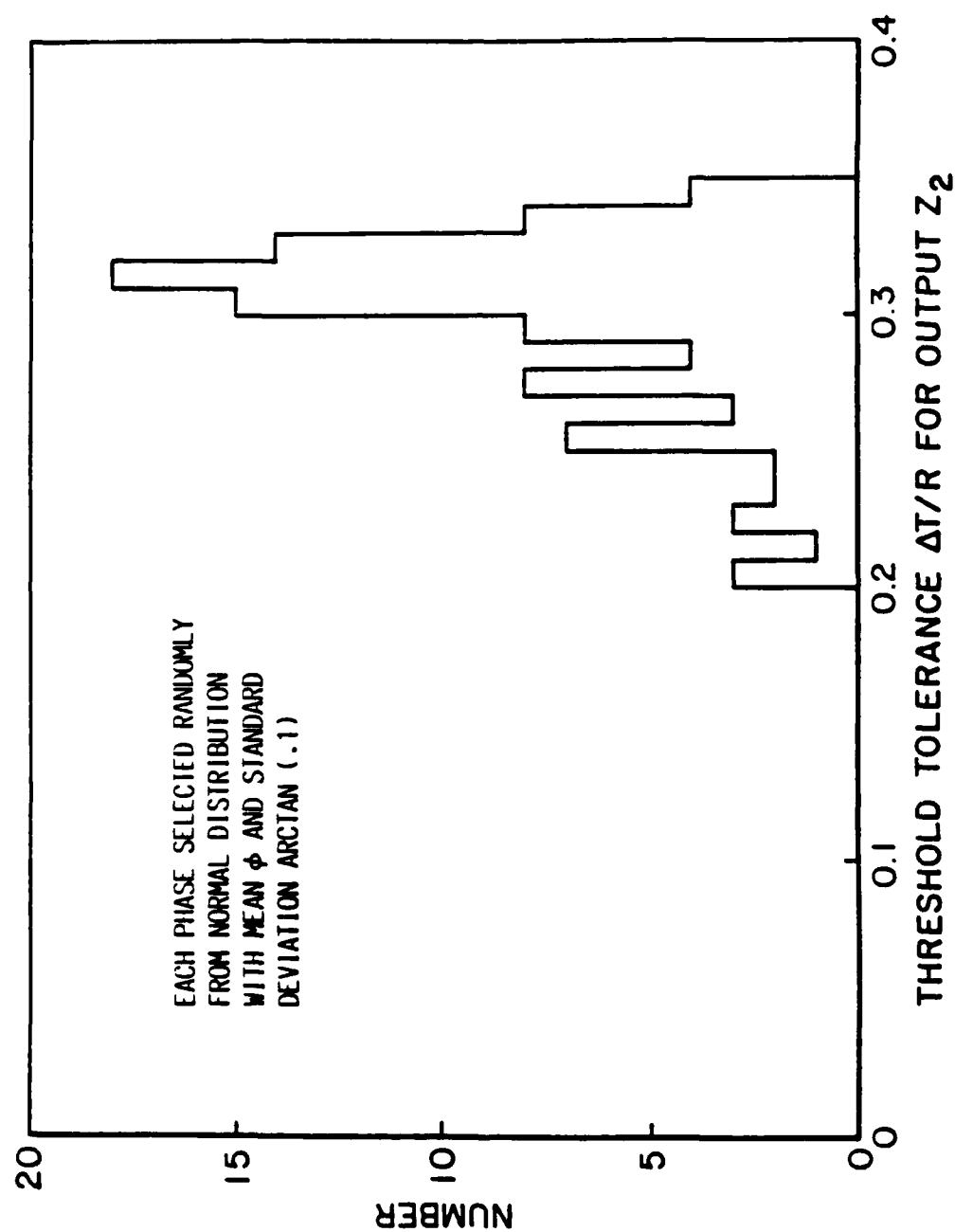


Figure 4. Histogram of threshold tolerance for output  $z_2$ .



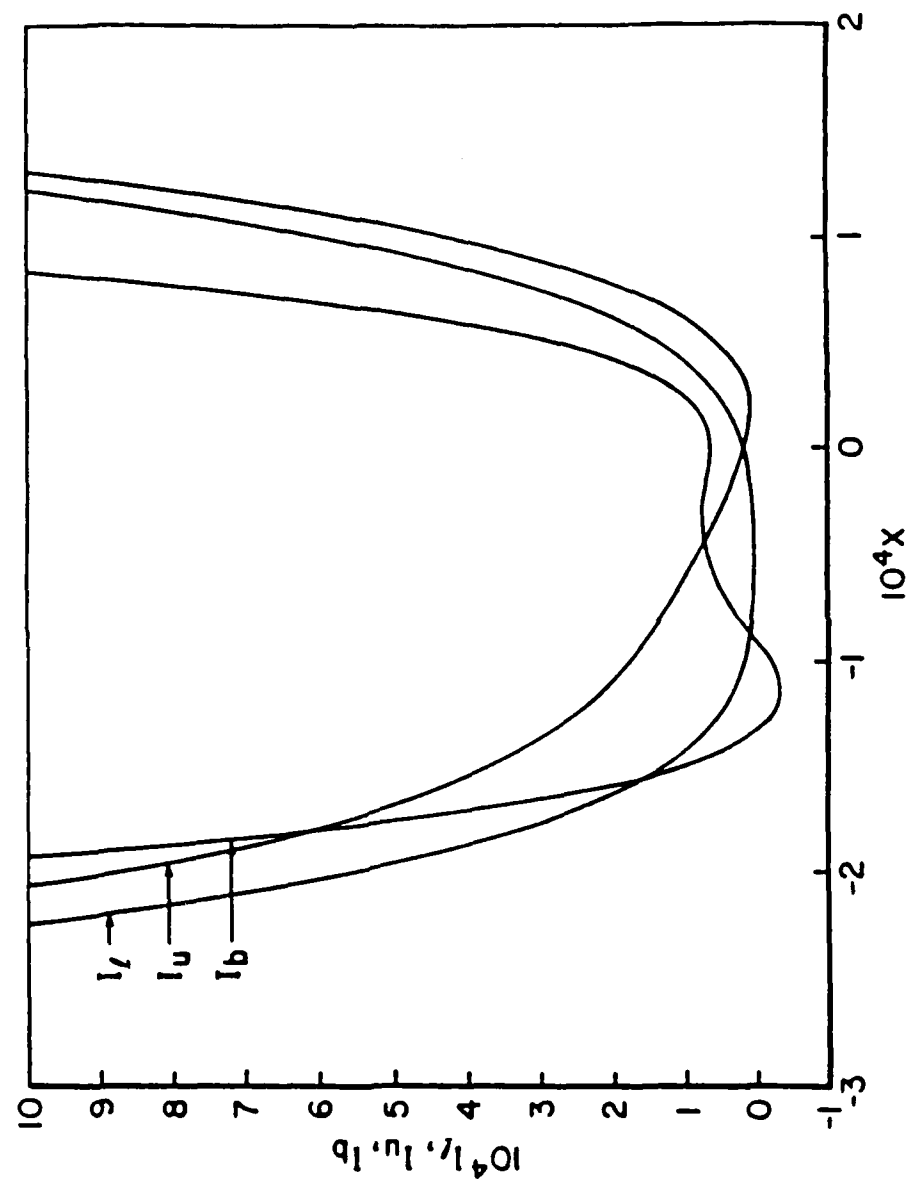


Figure 6. Graphs of approximate expressions for  $I_f$ ,  $I_u$ , and  $I_b$  versus  $x$  for  $\lambda = 628$  nm,  $b = 10$   $\mu$ m,  $s = 10$  cm, and  $\eta = .004$ .

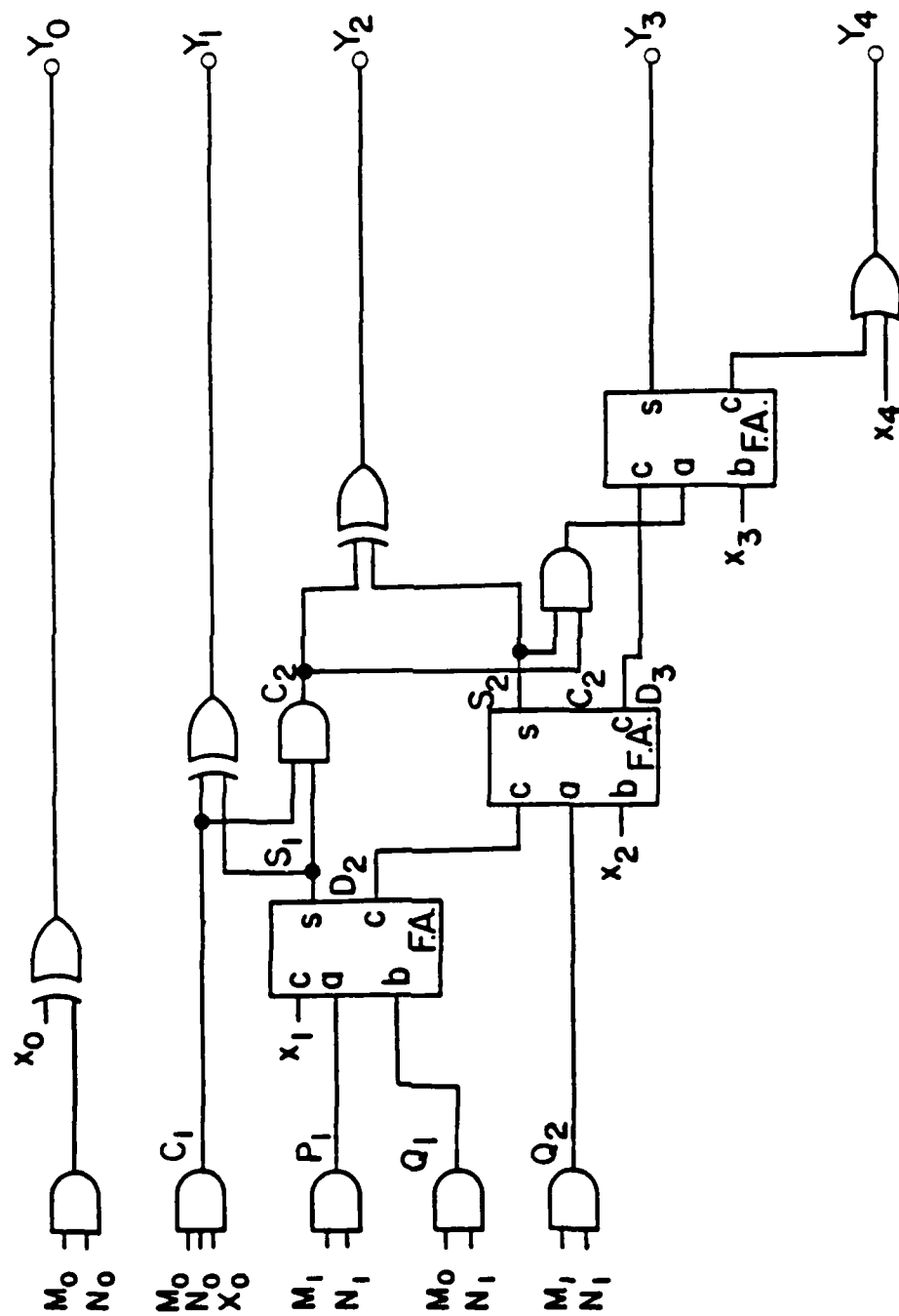


Figure 7. Conventional Boolean logic  $(2 \times 2 + 5 \times 5)$ -bit multiplier-adder. The full-adders (F.A.) require two logic levels and seven gates.

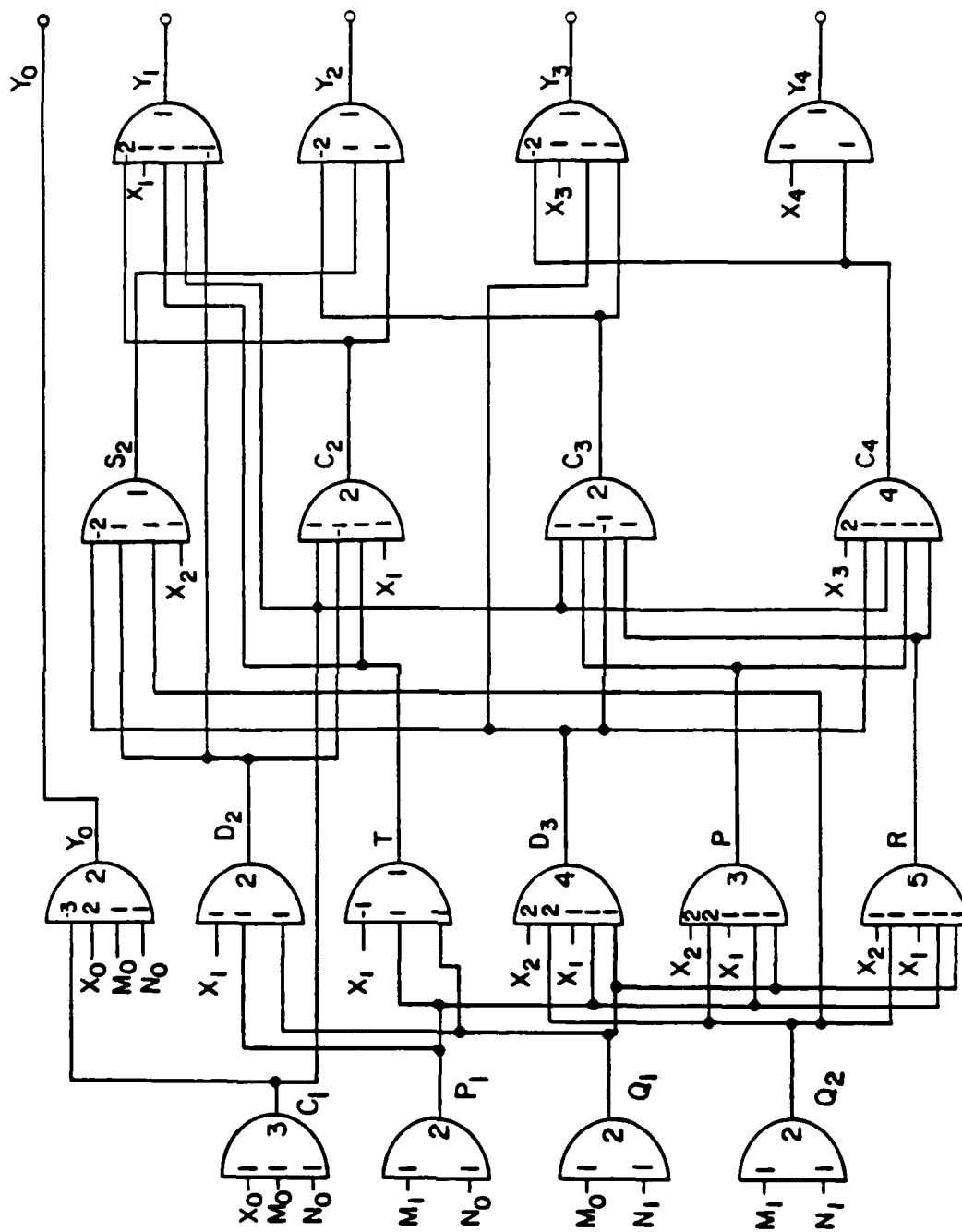


Figure 8. Threshold logic  $(2 \times 2 + 5 \cdot 5)$ -bit multiplier-adder.

	<u>CONVENTIONAL LOGIC</u>	<u>THRESHOLD LOGIC</u>
LOGIC LEVELS	15 (9)	9 (5)
LOGIC ELEMENTS	805 (38)	289 (18)
INTERCONNECTIONS	2141 (116)	1175 (70)

Figure 9. Comparison of conventional and threshold logic designs for a  $(8 \times 8 + 21 \rightarrow 21)$ -bit multiplier-adder. Figures for a  $(2 \times 2 + 5 \rightarrow 5)$ -bit multiplier-adder are in parenthesis.

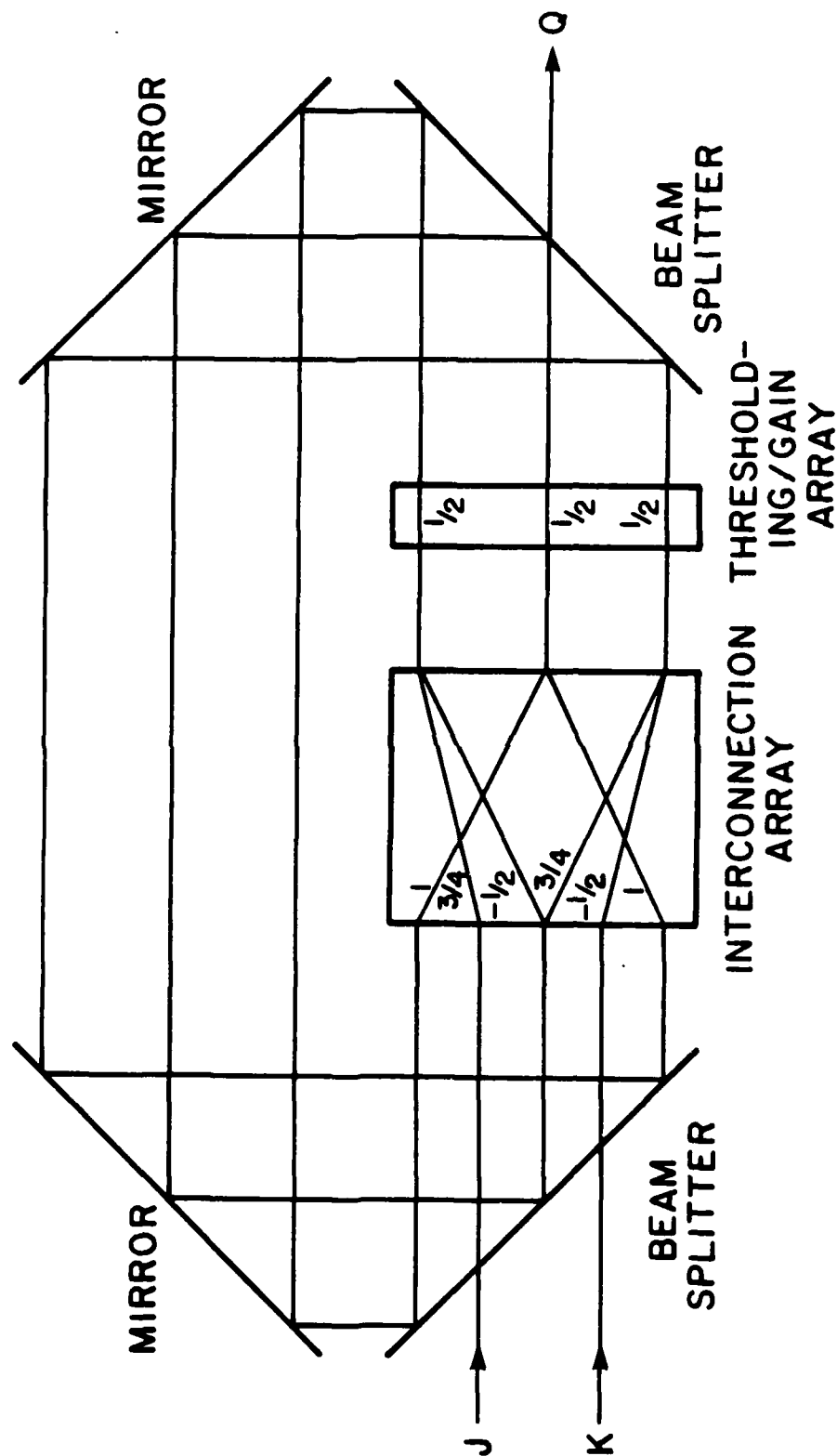


Figure 10. Internal thresholding design for a J-K flip-flop. Weights are indicated in interconnection array; thresholds are indicated in thresholding/gain array.

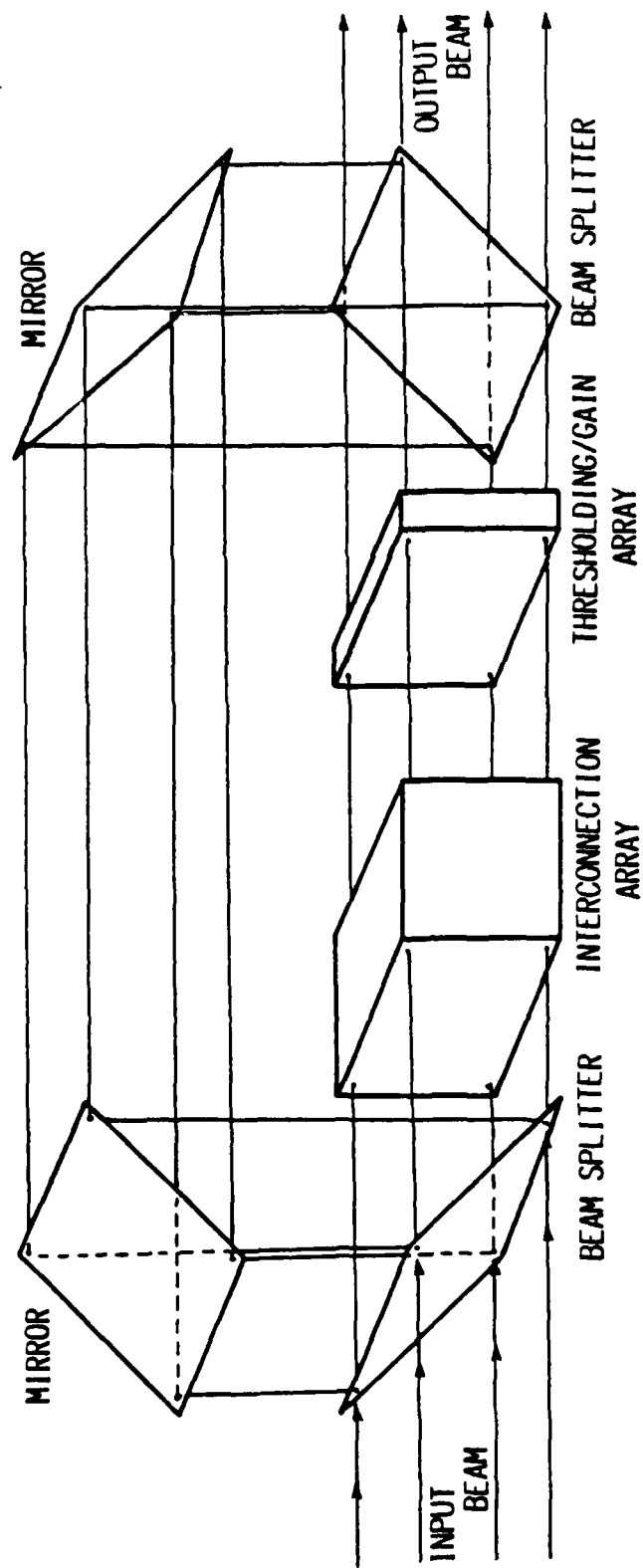


Figure 11. A general "femtosecond pancake" internal thresholding architecture for optical computing.



Image coding using  
pseudorandom shift register sequences

Steven C. Gustafson  
University of Dayton Research Institute  
Dayton, Ohio 45469

Abstract

The coding of digital images using linear feedback shift register (LFSR) sequences is considered analytically and in computer simulations. It is shown that LFSR sequences can provide efficient representations of binary and multilevel images in terms of a relatively small set of integers related (in limiting cases) to image complexity and randomness.

Introduction

Pseudorandom linear feedback shift register (LFSR) sequences are commonly employed in the characterization of one-dimensional communication signals, but no extensive use has been made of these sequences for characterizing images. This brief paper shows that in some situations LFSR sequences can provide efficient representations of binary and multilevel images in terms of a relatively small set of integers related (in limiting cases) to image complexity and randomness. The basic concept for a binary image is as follows: The image is coded into a binary sequence using the MacWilliams-Sloane<sup>1</sup> or other pseudorandomness-preserving construction. The Berlekamp-Massey algorithm<sup>2</sup> is then used to determine the lowest-order (i.e., smallest number of stages) LFSR that could have generated the coded sequence. This determination provides the coefficients in a two-element-field autoregression of the coded sequence or, equivalently, a set of positive integers that represent the generating LFSR feedback connection positions. In the limiting case where  $2^n - 1$  image pixels are characterized by an  $n$ -stage LFSR, the representation may have minimum complexity and maximum randomness. Some potential applications for this characterization are in the areas of data compression (e.g., texture coding) and coded aperture imaging<sup>4</sup> (e.g., determination of optimum aperture patterns). The limited objectives considered below include (1) finding the smallest-stage-number LFSR sequences that could have generated small random binary arrays, (2) obtaining histograms of stage number and period length, and (3) making observations on image coding using maximum-length and non-maximum length LFSRs.

Some maximum-length LFSR conditions and properties

Figure 1 is an example of a binary maximum-length LFSR. Here the contents of the second and third of the  $n = 3$  stages (represented by squares) are added modulo 2 (equivalent to an Exclusive-Or operation) and fed back to the first stage. As in any shift register, the contents of each stage are shifted to the right at each clock cycle. The LFSR then cycles through all  $2^n - 1$  non-zero states (e.g., the seven rows of stage contents indicated in Figure 1) before repeating, regardless of the initial state. The LFSR output is the contents of the last stage, and the LFSR is called maximum-length because its output pattern has the longest possible repeat period.

In general, the stages may be assigned  $q$  levels, where  $q$  is a prime number or a power of a prime. Three properties of the output of a general LFSR are necessary for a pseudorandom output:<sup>1</sup>

- (1) The pattern repeat period is  $q^n - 1$  output values,  $q^n - 1$  of which are zero and  $q^{n-1} - 1$  of which are nonzero.
- (2) A window of  $n$  consecutive stages located at all different positions on the output sequence will contain each of the  $q^n - 1$  nonzero  $n$ -tuples once.
- (3) The autocorrelation function of the output sequence is

$$\rho(t) = \begin{cases} 1, & t = 0 \\ -1/n, & q = 2, t \neq 0 \ (1 \leq t \leq 2^n - 2) \\ -1/(q^n - 1), & q \neq 2, t \neq 0 \ (1 \leq t \leq q^n - 2) \end{cases}$$

### MacWilliams-Sloane pseudorandom array construction

A pseudorandom array construction that preserves the above three properties in two dimensions is as follows:

- (1) Select a maximum-length LFSR sequence with  $n = 4, 6, 8, \dots$  stages.
- (2) Fill a  $2^{n/1} - 1$  by  $2^{n/2} + 1$  array with this sequence on the main diagonal using contiguous-array continuation. Figure 2 shows the placement of sequence positions 0, 1, 2, ... on a  $3 \times 5$  array ( $n = 4$ ).

Note that step 2 may be used to construct an array from a sequence or vice versa even if the sequence is not from a maximum-length LFSR. However, in this case the corresponding array will not in general have the three necessary properties of pseudorandomness.

### Berlekamp-Massey algorithm

The Berlekamp-Massey algorithm efficiently identifies the LFSR with the least number of stages that could have generated a given data sequence. It requires  $2n$  error-free consecutive sequence values to identify an  $n$ -stage maximum-length LFSR. It may be viewed as an elegant procedure for solving  $n$  auto-regression equations in  $n$  unknowns on a  $q$ -element field, where  $q$  is finite or infinite. The algorithm involves on the order of  $2n^2$  arithmetic operations and is presented in outline in Figure 3.

### Example image coding stage number and period length histograms

Figure 4 shows two example histograms of LFSR stage number for random bits that could be placed in a  $3 \times 5$  array and read as sequences using the MacWilliams-Sloane construction. The minimum stage number was determined by the Berlekamp-Massey algorithm for two situations. In the first, "Stage number for one case," the algorithm was applied to sequences read as shown in Figure 2 from 300 random binary arrays. (Bits in these arrays were assigned by a "good" computer random number generator.<sup>7</sup>) In the second, "Minimum stage number for 60 cases," the algorithm was applied to sequences read 60 times from each of 300 random binary arrays, and the minimum stage number was selected. Here the 60 cases were defined by the four possible array diagonal directions and by the 15 possible array starting positions. In the first situation the  $3 \times 5$  array required a mean of  $m = 7.80 \pm 1.05$  stages for LFSR representation. The corresponding image coding is specified by  $2m$  bits ( $m$  to specify the LFSR top connections and  $m$  to specify the LFSR initial loading), and thus may not be preferred to direct specification of the 15 image bits. In the second situation the  $3 \times 5$  array required a mean of  $6.31 \pm 0.87$  bits for LFSR representation. The corresponding image coding may thus be advantageous compared to direct image bit specification.

Figure 5 shows histograms of LFSR period length for random bits that could be placed in a  $3 \times 5$  array and read as sequences using the MacWilliams-Sloane construction. The period lengths are for minimum-stage-number LFSRs determined as in the "One case" situation in Figure 4 but for specified numbers of ones and zeros in the arrays. Note that the period lengths cover more than a factor of 1000 but that they have a modal value of  $2^7 - 1 = 127$ . This is expected because a sequence of 15 bits can be represented, in the absence of singularities, by an autoregression that yields eight equations in seven unknowns (corresponding to seven LFSR stages). Note also that arrays balanced with nearly equal numbers of ones and zeros have fewer extreme period lengths than do unbalanced arrays.

### Observations on image coding using LFSR sequences

Image coding using LFSR sequences divides naturally into two cases: coding using maximum-length LFSR sequences and coding using non-maximum-length LFSRs. In the maximum-length case, which may be useful, for example, in synthetic texture coding,  $q^n - 1$   $q$ -nary pixels may be represented by  $2n$   $q$ -nary values ( $n$  feed-back connections and  $n$  initial loadings). Thus, if  $n = 16$  and  $q = 2$ , 65535 pixels may be represented by only 32 binary values--a potentially great advantage. Maximum-length coding is also characterized by minimum complexity (minimum number of LFSR stages<sup>4</sup>) and maximum randomness (the three properties necessary for pseudorandomness and realized). In the non-maximum-length case, however, only limited data compression may be possible: Figure 4 indicates that 20% may be achieved on the average for random  $3 \times 5$  binary arrays. It may be concluded that maximum-length LFSR sequences offer large potential data representation or compression advantages and that these advantages may be realized in coding random or apparently random images.

### REFERENCES

1. F. J. MacWilliams and N.J.A. Sloane, "Pseudo-Random Sequences and Arrays," Proc. IEEE, **64**, pp. 1715-1729, Dec. 1976.

2. J. L. Massey, "Shift-Register Synthesis and BCH Decoding," IEEE Trans. Information Theory, IT-15, pp. 122-127, June 1969.
3. E. L. Key, "An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators," IEEE Trans. Information Theory, IT-22, pp. 732-736, Nov. 1976.
4. T. M. Cannon and E. E. Fenimore, "Coded Aperture Imaging: Many Holes Make Light Work," Opt. Eng. 19, pp. 283-289, May-June 1980.
5. R. E. Blahut, "A Universal Reed-Solomon Decoder," IBM J. Res. Devel., 28, pp. 150-158, Mar. 1984.
6. K. Y. Lin, "Architecture for VLSI Design of Reed-Solomon Decoders," IEEE Trans. Computers, C-33, pp. 177-189, Feb. 1984.
7. D. E. Knuth, Seminumerical Algorithms, 2nd ed., Addison-Wesley, NY, 1981.
8. M. Spann, T. Dodgson, and R. Wilson, "Coding Textures," 2nd Intl. Symp. Optical and Electro-Optical App. Sci. and Eng., Cannes, France, 5 Dec 1985.

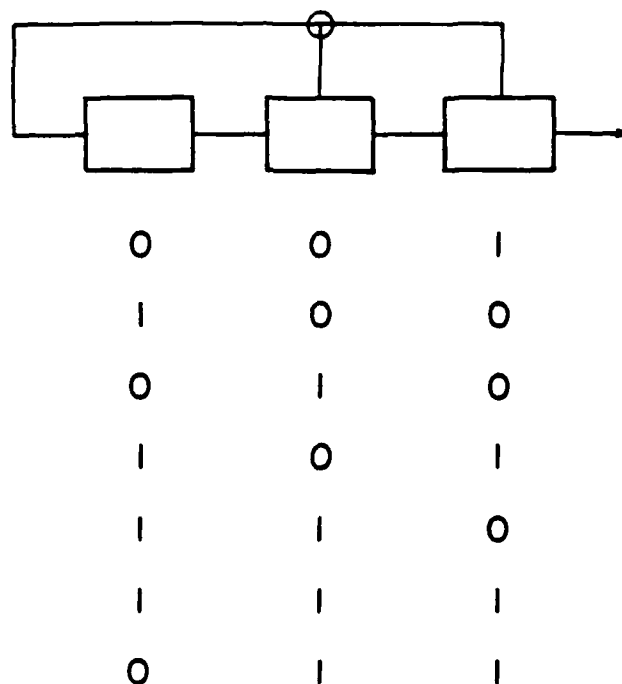


Figure 1. Example of a linear feedback shift register (LFSR):  
n = 3 stage, maximum length.

	0	6	12	3	9
	10	1	7	13	4
	5	11	2	8	14

Figure 2. MacWilliams-Sloane pseudorandom array construction.

- Data

$$S_0, S_1, \dots, S_{2N-1}$$

- Definitions

$$C_p(D) = 1 + C_{p,1}D + C_{p,2}D^2 + \dots + C_{p,i}D^i$$

$$d_p = S_p + C_{p,1} S_{p-1} + C_{p,2} S_{p-2} + \dots + C_{p,p} S_0$$

- Expressions

$$C_{n+1}(D) = C_n(D) - \frac{d_n}{d_{k_n}} D^{n-k_n} C_{k_n}(D)$$

$$i_{n+1} = \begin{cases} i_n, d_n = 0 \\ \max(i_n, n - k_n + i_{k_n}), d_n \neq 0 \end{cases}$$

- Initialization

$$S_n = S_0 \quad i_0 = i_{-1} = 0$$

$$C_0(D) = C_{-1}(D) = 1 \quad k_0 = -1$$

$$d_0 = d_{-1} = 1$$

Figure 3. Outline of Berlekamp-Massey algorithm.

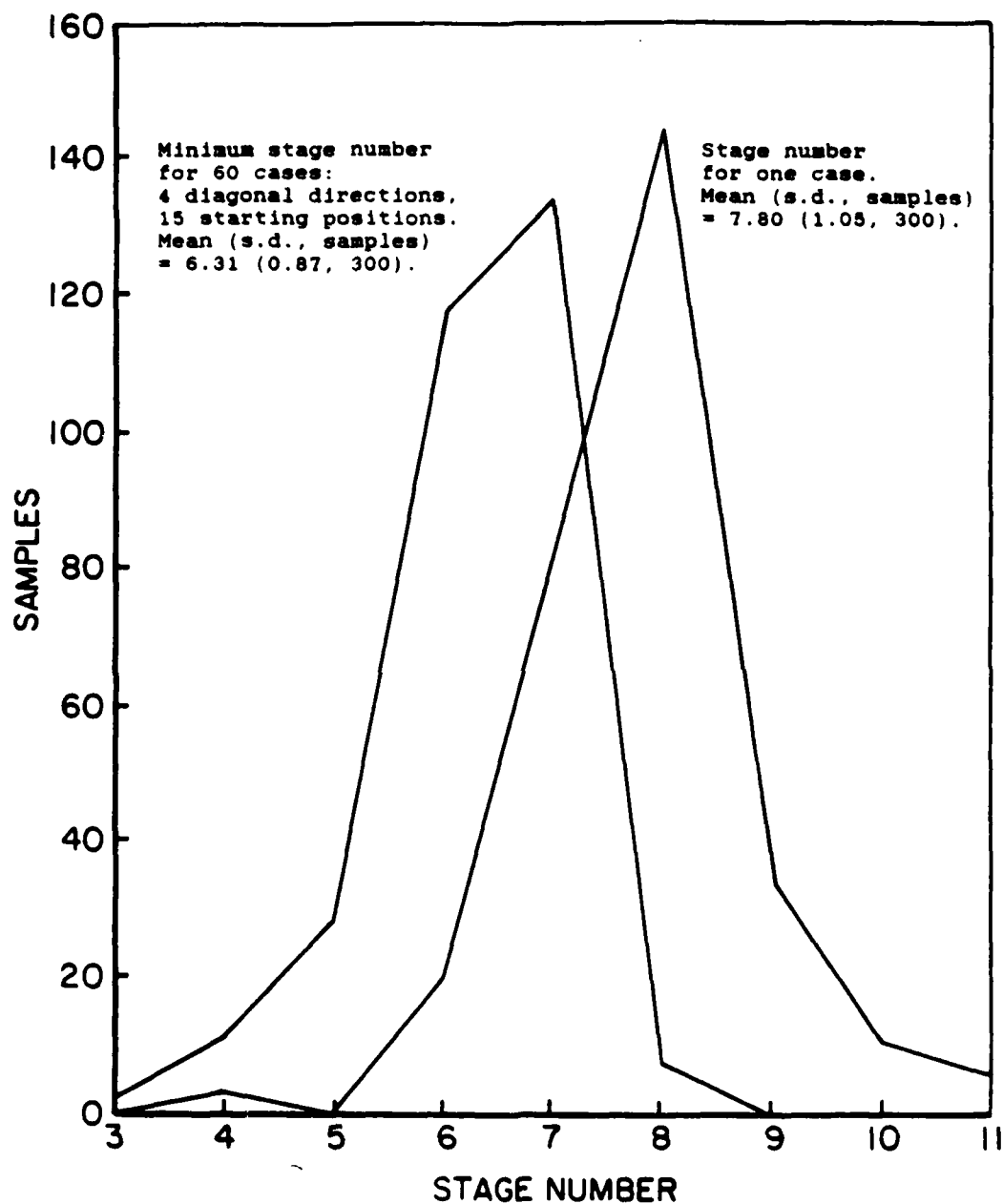


Figure 4. Histograms of LFSR stage number: random bits in a 3-x-5 array; sequence read using MacWilliams-Sloane construction.

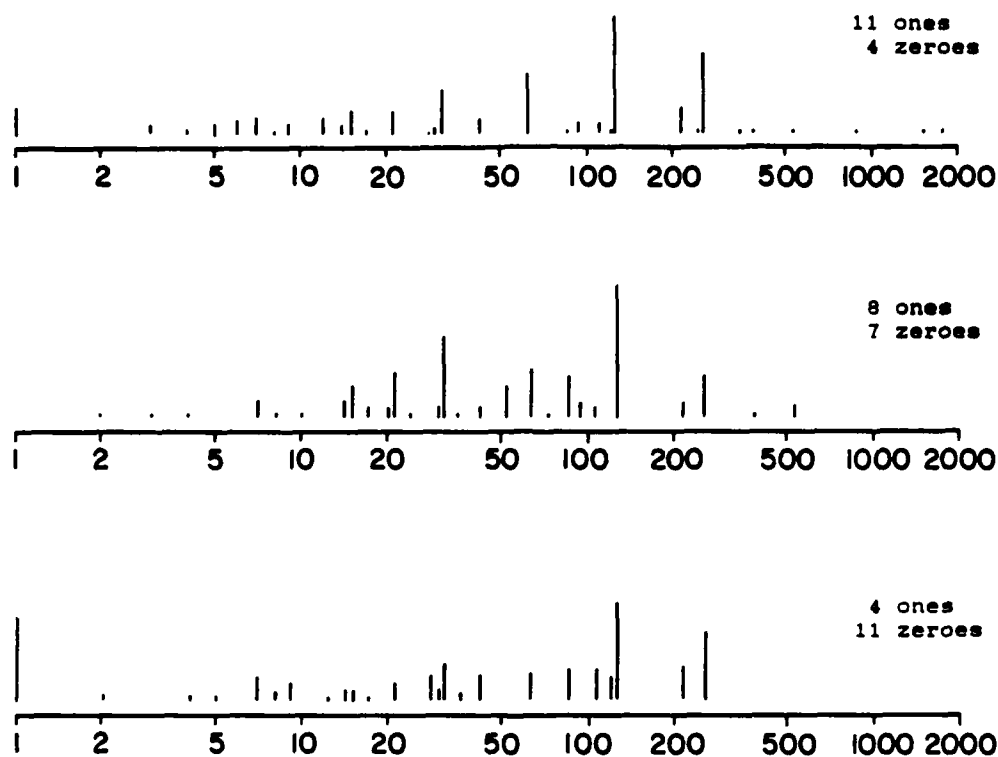


Figure 5. Histograms of LFSR period length: random bits in a 3-x-5 array; specified numbers of ones and zeroes.

# Analysis of Threshold Logic for Applications to Optical Computing

Steven C. Gustafson, David L. Flannery

University of Dayton Research Institute,  
300 College Park Ave., Dayton, Ohio 45469

and

Robert O. Winder  
24 Deer Path  
Princeton, N.J. 08540

## Abstract

Optical threshold logic implementations of register-level operations such as multiply-accumulate are considered. Specific 2- and 8-bit multiply-add designs using both conventional (Boolean) and threshold logic are described and compared. The threshold logic designs are shown to have advantages of factors of 2 or 3 in number of logic levels, number of logic elements, and number of interconnections. The possibility of all-optical implementations (optical logic elements and optical internal connections) of these designs is discussed in terms of integrated optical networks and nonlinear optical devices, particularly in GaAs structures.

## Introduction

Optics-based digital processing may be carried out on the processor, register, or gate level. Many processor-level systems have been described, including systolic optical array processors that make effective use of acousto-optic cells, lens assemblies, and optical source and detector arrays, to perform digital matrix-vector multiplication or other matrix algebra operations.<sup>1,2,3</sup> Numerous optics-based gate-level devices have also been considered, particularly devices based on optical bistability<sup>4,5</sup> that could be very high-speed replacements for conventional all-electronic logic gates. Optics-based register level modules such as scalar multipliers, multiply-accumulators, correlators, etc., have, with some exceptions,<sup>6,7,8,9</sup> received less attention. Desirable features of optically implemented register-level architectures include modularity (for fault-tolerance and application flexibility) and either electronic or optical inputs and outputs (for compatibility with other components or subsystems).

Since optical-electronic (or electronic-optical) conversions are generally costly in terms of speed, power consumption, device size, etc., it is anticipated that optics-based register-level modules will require all-optical (or nearly all-optical) internal connections in order to realize a potential for substantial performance improvements compared to all-electronic designs. Such optics-based modules need not make use of the same Boolean logic gate structures (OR, NAND, etc.) that have become conventional in all-electronic integrated circuit technology. In particular, threshold logic constitutes a more general technology that includes conventional Boolean logic as a special case and that requires no more and often significantly fewer logic levels, elements, and interconnections to carry out the same function. Threshold logic elements and networks may have effective integrated (or near-integrated) optical implementations involving nonlinear optical (e.g., bistable) devices. Some specific multiply-add threshold logic networks are considered in this paper, and potential all-optical implementations are discussed.

## Threshold Logic Elements

A threshold logic element, in the form usually considered in the literature,<sup>10,11,12</sup> has several binary inputs and one binary output. It executes a logic function of the inputs that depends on the sum of products obtained by multiplying each binary input by a fixed analog weight. If this sum is less than a fixed threshold level, the element output is zero; otherwise the output is a one. For example, an element with binary inputs  $x_1$ ,  $x_2$ ,  $x_3$ , and  $x_4$  and corresponding weights  $w_1 = 2$ ,  $w_2 = 1$ ,  $w_3 = 3$  and  $w_4 = 1$  with a threshold level  $T$  in the range  $4 < T \leq 5$  implements the logic function  $y = x_1x_2x_3x_4 + x_1x_3$ , where  $y$  is the binary output, the implied product is an AND operation, the plus sign is an OR operation, and the superscript bar is a NOT operation. Note that the implementation of this function using conventional logic requires three logic levels (including inversion), whereas the threshold logic element constitutes a single logic level and thus has a potential factor-of-three speed advantage. Note also that in this example there may be up to an 11% variation in the value of  $T$  without affecting element operation, that similar tolerances will apply to the analog weights, and that these generally non-precise analog weighting and thresholding operations may be viewed as intuitively appropriate for optical implementation.



Characteristics such as fewer logic levels, practical analog weight and threshold tolerances, and possible optical implementations may apply to threshold logic elements in general and to suitably designed networks of such elements.

### Threshold Logic Networks

Conventional Boolean logic and threshold logic designs may be compared for specific register-level operations: One relatively simple example is a two-bit multiply-add module. This module multiplies two two-bit input numbers M and N, adds the result to a five-bit input number X, and outputs the result as a five-bit number Y. Such a module could be the heart of a clocked multiply-add module in which input and output registers and timing control would be included. (For examples of existing digital electronic chips with this function, see "The VLSI Data Book," pp. 231-294, TRW, La Jolla, CA, 1984.) In clocked operation, the output Y could be fed back to the input X to achieve a multiply-accumulate operation with the capability of accumulating up to three products without overflow.

A conventional Boolean logic design for a two-bit multiply-add module is diagrammed in Figure 1, where the subscripts on M, N, X, and Y designate binary number position ( $2^0$ ,  $2^1$ , etc.). For simplicity, two well known and frequently occurring multigate configurations have been grouped and represented by single symbols. The exclusive OR (XOR) function may be accomplished in two logic levels using two AND gates, one OR gate, and two inverters. The full-adder is a standard functional module which can be cascaded to form a ripple-carry adder for two numbers of any bit length. The three inputs, a, b, and c, represent the two corresponding-order bits of the two input numbers and the carry bit, which is obtained (normally) from the carry output of the previous adder. The output s is the corresponding bit of the sum, and the output c is the carry bit normally fed to the next adder. A minimized two-level logic configuration which performs the full-adder function requires five AND gates, two OR gates, and four inverters.

A threshold logic design for a two-bit multiply-add module<sup>13</sup> is diagrammed in Figure 2 and is composed entirely of threshold logic elements, with fan-in and fan-out limited to five. Weights are indicated inside the element symbol adjacent to the element input lines, and the element threshold is indicated adjacent to the output line.

Note that the Boolean logic design in Figure 1 requires a total of 38 logic gates and 18 inverters. Inverters are not normally included in the logic level count; however, they do require circuitry and space. This design has a maximum propagation path of 9 logic levels. The threshold logic design in Figure 2 requires 18 threshold logic elements and involves only 5 logic levels. It is also meaningful to compare the number of interconnect lines required by the designs. This count is 116 for the Boolean logic design versus 70 for the threshold logic design. The comparison may thus be summarized by stating that the threshold logic design is superior by factors of roughly two with regard to number of logic levels, gate count, and number of interconnections. Although the design example analyzed is of questionable practical significance, the comparison strongly favors threshold logic.

A similar design comparison was carried out for what is perhaps a more practically useful case: an 8-bit multiplier-adder that multiplies two 8-bit numbers, adds a 21-bit number, and outputs a 21-bit result. In this case, the fan-in and fan-out constraints were increased to eight. The results of this design<sup>13</sup> (with the 2-bit multiplier-adder results in parenthesis) are summarized in Figure 3. As can be seen, the threshold logic advantage in gate count ratio has increased to almost three-to-one, while the logic level and interconnection ratios have remained at about two-to-one. This design is reasonably complex, and one is tempted to conclude that the results are indicative of what may be obtained for more general complex designs. In any case, the results indicate a significant advantage for at least one design of practical importance. If viewed in terms of the ratio of processing speed to power consumption, it can be argued that the results also indicate an advantage considerably greater than just the separate level, element, or interconnection ratios, since these ratios each contribute to either increased processing speed or reduced power consumption or both.

### Optical Threshold Logic Implementation

As indicated above, it is anticipated that all-optical elements and internal connections will be required for register-level modules that realize their potential for commanding and enduring performance advantages compared to all-electronic designs. If size advantages are also to be realized, integrated optical technology appears to constitute the only viable approach. Furthermore, among the many material systems that have been investigated in this technology (LiNbO<sub>3</sub>, glass/SiO<sub>2</sub>/Si, etc.), only GaAs/GaAlAs systems have a strong potential for the complete integration of optical sources and detectors on one substrate. Recently, enhanced electro-optic effects reported in GaAs/GaAlAs multiple quantum well (MQW) structures<sup>14,15,16</sup> have opened up the prospect of the eventual integration of high-performance optical gates in dense arrays on GaAs integrated optical structures. (Other

material systems are being pursued along similar lines, including a notable European effort centered on InSb.<sup>17,18)</sup>

The thresholding function for TL elements would ideally be carried out using nonlinear optical methods, particularly methods related to optical bistability. For such methods an input intensity below a threshold level yields a small output intensity while an input intensity above the threshold level yields a large output intensity. Recent work<sup>19</sup> indicates that the required material nonlinearities for these methods are two to four orders of magnitude beyond the values that characterize current uniform electro-optic materials, such as LiNbO<sub>3</sub> or GaAs for the case where the optical input is provided by laser diode sources. However, MQW structures and certain "exotic" materials such as multiple-layer Langmuir-Blodgett organic films<sup>20</sup> may have the required nonlinearity properties. In particular, a variety of single element MQW electro-optic devices operating at room temperature with promising speed and energy parameters have recently been reported.<sup>14</sup> These devices are relatively simple structures which have good potential to be integrated in large arrays of small area devices, i.e., in GaAs integrated optical circuits. A key feature is that this and several other such MQW GaAs structures have optical inputs and outputs, thus supporting the concept of the all-optical modules that use optical logic elements and optical internal connections exclusively.

As a basis for a discussion of the potential of register-level optical threshold logic elements and networks, consider the all-electronic 16-bit multiplier chip which has been announced by Fujitsu.<sup>21</sup> This GaAs chip performs a 16-bit multiply in 10.5 ns, consumes approximately 0.3 W, contains about 3168 conventional logic gates, and has roughly 18 logic levels if 0.6 ns per level is assumed. Thus, the chip performs at a rate of 95 Megaoperations per second (MOPS) with a specific power consumption of about 300 MOPS/Watt. As a very crude projection, it may be assumed that a factor-of-three advantage in number of logic elements and no advantage in logic levels would be realized in a threshold logic 16-bit multiplier design. Thus about 1,100 optically connected threshold logic elements would be required. Recent results<sup>14</sup> demonstrated switching energies of 4 to 20 FemtoJoules per square micron for GaAs MQW gates. Based on an assumed-switching time of 100 ps, the time for a 16-bit multiply is  $16 \times 100 \text{ ps} = 1.6 \text{ ns}$ . Assuming an eight square micron gate area, the power dissipation for the threshold logic module may be as small as 1,100 elements  $\times 32 \text{ FemtoJoules}/1.6 \text{ ns} = 20 \text{ mW}$ . If the size of a threshold logic element is estimated to be 50 microns by 100 microns (including an allowance for interconnections), then the area required by 1,100 such elements is  $5.5 \text{ mm}^2$  which is comparable to the area of typical current IC chips. The thruput of the projected optical module is about 500 MOPS, but the optical module has a specific power consumption of about 25,000 MOPS/Watt—two orders of magnitude better than the all-electronic chip.

#### References

1. "Optical Computing," Opt. Eng. Special Issue, Jan. 1984.
2. "Optical Computing," Proc. IEEE Special Issue, Jan. 1984.
3. "Optical Computing," Opt. Eng. Special Issue, Jan. 1985.
4. Gibbs, H. M., S. L. McCall, T. N. C. Venkatesan, "Optical Distable Devices: The Basic Components of All-Optical Systems?", Opt. Eng. 19, pp. 463-468, 1980.
5. Abraham, E. C. T. Seafon, and S. D. Smith, "The Optical Computer," Sci. Am., pp. 85-93, Feb. 1983.
6. Tai, A., I. Cindrich, J. R. Fienup, and C. C. Aleksott, "Optical Residue Arithmetic Computer with Programmable Computation Modules," Appl. Opt. 18, pp. 2812-2823, 1979.
7. Polky, J. N. and D. D. Miller, "Optical Waveguide Circuit Design of an Adaptive Filter in the Residue Number System," Appl. Opt. 21, pp. 3539-3551, 1982.
8. Arrathoon, R., and M. H. Hassoun, "Optical Threshold Logic Elements for Digital Computation," Opt. Lett. 9, pp. 143-145, 1984.
9. Gustafson, S. C., "Electro-Optical Intensity Modulation of Near-Surface Light for Optical Signal Processing Applications," Proc. SPIE 517, pp. 163-166, 1984.
10. Muroga, S., Threshold Logic and Its Applications, Wiley, N.Y., 1971.

11. Lewis II, P. M., and C. L. Coates, Threshold Logic, Wiley, N.Y., 1967.
12. Dertouzos, M. L., Threshold Logic: A Synthesis Approach, MIT Press, Cambridge, MA, 1965.
13. Winder, R. O., Private communication, February 1985.
14. Miller, D., D. Chemla, T. Damen, T. Wood, C. Burrus, A. Gossard, W. Wiegmann, "Quantum Well Optical Modulators and Self Electro-Optic Effect Devices (SEED's)," Paper MC6-1, Optical Society of America Topical Meeting on Optical Computing (Incline Village, Nevada, March 1985).
15. Chemla, D. S., "Quantum Wells for Photonics," *Physics Today*, May, 1985.
16. Wood, T. H., C. Burrus, D. Miller, D. Chemla, T. Damen, A. Gossard, W. Wiegmann, "High-speed Optical Modulation with GaAs/GaAlAs Quantum Wells in a p-i-n Diode Structure," *Appl. Phys. Lett.* 44, (1), January 1984.
17. Kar, A., J. Mathew, S. Smith, B. Davis, and W. Prettl, "Optical Bistability in InSb at Room Temperature with Two-Photon Excitation," *Appl. Phys. Lett.* 42, p. 334, 1983.
18. Seaton, C., S. Smith, P. Tooley, M. Prise, and M. Taghizadeh, "Realization of an InSb Bistable Device as an Optical AND Gate and Its Use to Measure Carrier Recombination Times," *Appl. Phys. Lett.* 42, p. 131, 1983.
19. Verber, C., Private communication, March 1985.
20. Neff, J. A., "Future Directions in Optical Processing," NASA Conf. 2302, Hampton, VA, August 1983.
21. Eden, R. C., A. Livingston, B. Welch, "Integrated Circuits, the Case for Gallium Arsenide," *IEEE Spectrum* 20, 12, p. 30-36, December 1983.

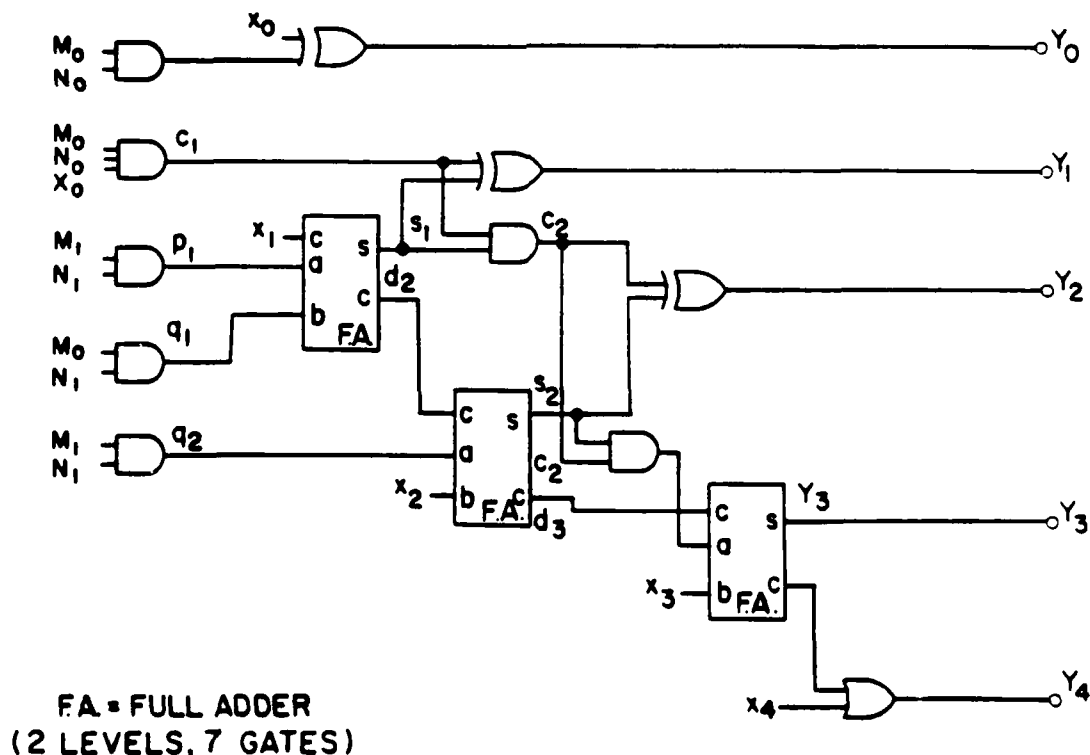


Figure 1. Conventional Boolean Logic  $(2 \times 2 + 5 - 5)$  -Bit Multiplier-Adder.

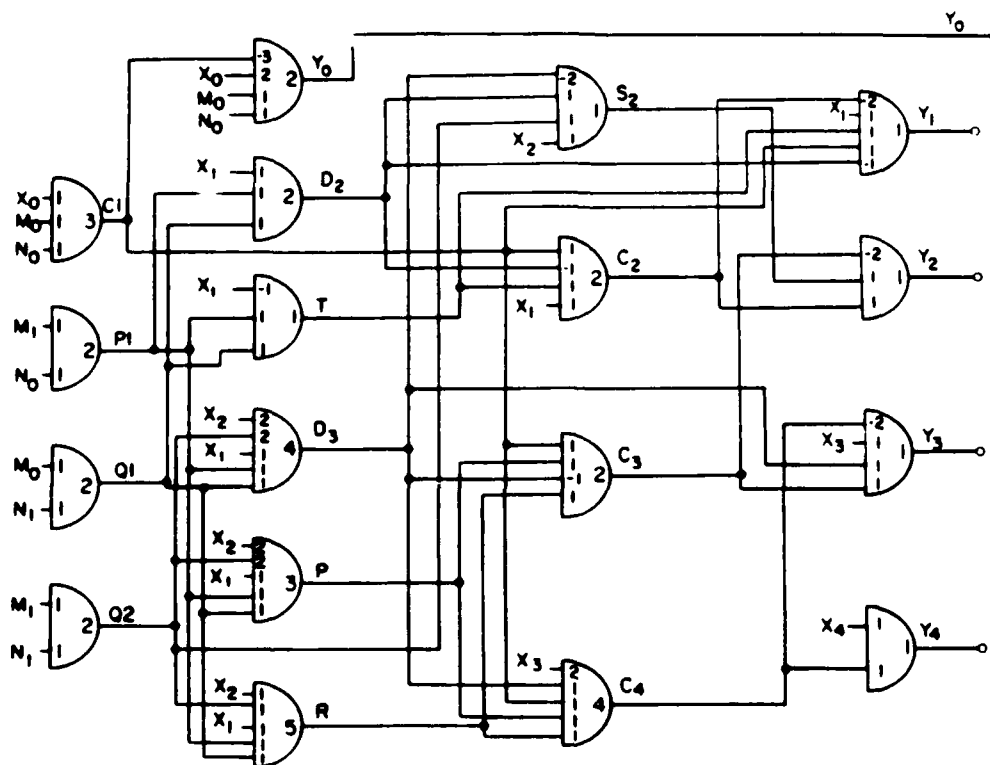


Figure 2. Threshold Logic  $(2 \times 2 + 5 - 5)$  -Bit Multiplier-Adder.

	CONVENTIONAL LOGIC		THRESHOLD LOGIC	
LOGIC LEVELS	15	(9)	9	(5)
LOGIC ELEMENTS	805	(38)	289	(18)
INTERCONNECTIONS	2141	(116)	1175	(70)

Figure 3. Comparison of Conventional and Threshold Logic Designs for a  $(8 \times 8 + 21 - 21)$  -Bit Multiplier-Adder.  $((2 \times 2 + 5 - 5)$  -Bit Multiplier-Adder Figures are in Parenthesis.)

## OPTICAL IMPLEMENTATIONS OF LUMPED THRESHOLD LOGIC

S. C. Gustafson, J. A. Kirk, and G. R. Little

University of Dayton Research Institute  
300 College Park, Dayton, OH 45469

and

R. P. Kenan and C. M. Verber

Battelle Columbus Laboratories  
505 King Ave., Columbus, OH 43201

### Abstract

Optically implemented threshold logic systems that are characterized by thresholding operations concentrated at one functional location are considered. The objective is to identify architectures and associated integrated optical and holographic techniques that might be used to design superior register-level computation modules. A complete design for a lumped threshold 2-bit multiplier is presented as an example, and methods for general lumped threshold module synthesis are discussed.

### Introduction

Threshold logic has received attention recently because it may provide significant performance advantages for a broad range of mathematical operations and because it may have efficient optical implementations.<sup>1,2</sup> This paper considers lumped threshold logic systems, which are defined as systems in which thresholding operations are concentrated at one functional location. The objective is to identify architectures and associated holographic and integrated optical techniques that might be used to design register-level computation modules, such as pure-radix multipliers, multiply accumulators, etc., with commanding and enduring advantages in speed, power consumption, size, fault-tolerance, etc., over current and projected all-electronic alternatives.

Figure 1 shows two general types of systems that relate sets of inputs and outputs using weighting and thresholding operations. Here "weighting" refers to interconnects with selected connection strengths, and "thresholding" refers to decisions based on inequality criteria. Distributed threshold logic systems generally have numerous distinct elements of the same type, each of which performs weighting and thresholding functions. For example, each element could conceivably be an optical-input-optical-output multiple quantum well (MQW) gate, and all elements could be optically interconnected so that the thresholding operations would be distributed throughout the system. In contrast, lumped threshold logic systems generally have only two functional units, one for weighting and one for thresholding. For example, the weighting operation could be accomplished by passive or active (i.e., programmable) integrated optical diffracting elements, and the thresholding operation could be accomplished by photo-detectors at an optical-to-electronic output interface. In this case the (nonlinear) thresholding operation is global or concentrated at the output of the system.

### Lumped Threshold 2-Bit Multiplier

The 2-bit multiplier may be used as a simple example of lumped threshold logic design. Figure 2 is a 2-bit multiplier truth table for the multiplication of binary numbers  $x_1 x_0$  and  $y_1 y_0$  to obtain  $z_3 z_2 z_1 z_0$ . Suppose that the four input bits are represented by 0 if they are zero and by  $x_1 = A_1 \exp(i\phi_1)$ ,  $x_0 = A_2 \exp(i\phi_2)$ ,  $y_1 = A_3 \exp(i\phi_3)$ , and  $y_0 = A_4 \exp(i\phi_4)$  if they are ones. If these expressions correspond to waves in a geometrical optics approximation where all source-source, source-detector, and detector-detector distances are large compared to the wavelength, then the 2-bit multiplier may be designed as shown in Figure 3a. Here  $x_0$ ,  $x_1$ ,  $y_0$ , and  $y_1$  are optical point sources,  $z_0$ ,  $z_1$ ,  $z_2$ , and  $z_3$  are point photodetectors. The lines indicate optical paths, each of which may have a selected attenuation and phase shift that might be implemented by a hologram or integrated optical diffracting elements.

The required attenuations and phase shifts may be obtained by solving sets of simultaneous nonlinear inequalities derived from the truth table. For example, the 12th row and the  $z_2$  column of the table imply a signal at photodetector  $z_2$  that must equal or exceed a threshold  $T_2$ :

$$|A_1 \exp(i\phi_1) + A_3 \exp(i\phi_3) + A_4 \exp(i\phi_4)|^2 \geq T_2 \quad (1)$$

Similar expressions may be obtained so that each of the four output columns (labeled  $z_3$ ,  $z_2$ ,  $z_1$ , and  $z_0$ ) is described by a set of 16 (one for each table row) simultaneous nonlinear inequalities in 9 unknowns: four amplitudes ( $A_1$ ,  $A_2$ ,  $A_3$ , and  $A_4$ ); four phases ( $\phi_1$ ,  $\phi_2$ ,  $\phi_3$ ,  $\phi_4$ ); and one threshold ( $T_1$ ,  $T_2$ ,  $T_3$ , or  $T_4$ ). Solutions are to be found for each of these four overdetermined inequality sets (which involve terms such as  $A_1^2$ ,  $2A_1A_2\cos(\phi_1-\phi_2)$ , etc.) such that the amplitudes, phases, and thresholds obtained all have acceptable tolerances or ranges over which they may vary without affecting proper 2-bit multiplier operation.

One solution which involves only phase shifts (no attenuations) and which may have practical tolerances is given in Figure 3b, where the  $\phi$  column gives the phase shifts required for each of the four paths (in order) to each detector, the  $T$  column gives the threshold value for each detector when  $A_1 = A_2 = A_3 = A_4 = 1$ , and the  $\Delta T/R$  column gives the fraction of the total signal range on each detector over which its threshold may vary. Figure 4 is a histogram of  $\Delta T/R$  for output  $z_2$  generated by selecting each of the four phases for this output randomly from normal distributions with means at their design values and standard deviations equal to  $\arctan(.1)$ . These standard deviations correspond to 10% displacement of the phase vectors, and Figure 4 shows that such variations reduce the threshold tolerance  $\Delta T/R$  for output  $z_2$  from 37% to about 20%. Similar acceptable tolerances may be obtained for the other outputs and should be amenable to engineering design.

#### Optical Implementations

The 2-bit multiplier design described above is based on the ability of optics to provide noninterfering interconnections<sup>3</sup> which (1) are parallel in that interconnection time is essentially independent of interconnection length or weight and which (2) lead to system operation times essentially limited only by the response times of sources or detectors. These interconnections may be provided by passive diffracting elements in the form of an ordinary or bulk thin or thick film hologram in which light propagates approximately normal to the hologram plane. These interconnections may also be provided in integrated optical implementations by passive diffracting elements formed on or near a substrate surface such that light propagates approximately parallel to the surface. Such integrated optical implementations could use surface relief or photorefractive mechanisms to form the diffracting elements on GaAs, LiNbO<sub>3</sub>, glass or other substrates.

Integrated optics has potential for implementing lumped threshold computation modules with superior advantages in size, power consumption, reliability, etc. This technology also has potential for implementing real-time programmable interconnections or weightings using electro-optically modulated diffracting element structures (or, ultimately, all-optical nonlinear devices). This capability would be important, for example for neural network architectures that could perform "intelligent" adaptive and symbolic processing.<sup>4</sup> Figure 5a shows a direct implementation of programmable interconnections using a segmented array electro-optic grating.<sup>5</sup> Here the interconnection shown in Figure 3a are reordered so that no connection paths cross by providing a uniform optical input and applying one of two voltages to the grating segments in accordance with the input bits  $x_1$ ,  $x_0$ ,  $y_1$ , and  $y_0$ . Note that individual grating segments  $G_1$  are tilted at angles  $\theta_1$  so that parallel input beams  $\psi_1$  are directed to a detector with threshold  $T$  after weight  $W_1$  is applied as determined by programmable voltages  $V_1$ . However, since there must be a minimum deflection angle, the lateral separation  $D$  must become large as the number of segments  $N$  increases. Figure 5b shows one way of circumventing this problem using an integrated optical lens, which also permits (1) a common angle  $\phi$  for all grating segments and (2) the elimination at a stop (or monitoring) of undiffracted light. Figure 6 shows an integrated electro-optical channel-guide implementation. Here the channels are addressed through horns, and phase modulation is provided by surface electrodes. The output horns terminate in multimode regions whose outputs are plane waves which impinge on a surface grating at the Bragg angle. The contributions from the individual horns mix in the grating and produce an output which is a function of the phase differences between all pairs of input beams. Advantages of this arrangement are compactness and isolation of the detector from stray light.

Figure 7 shows how a hologram might be optically generated for implementing certain input-output relationships or truth tables (including the 2-bit multiplier truth table) in a lumped threshold system. One possibility, the Fourier transform hologram in Figure 7a, is multiply-recorded using object sources  $O$  related to output truth table elements and

reference sources  $R$  related to input truth table elements. Note that these sources need not be evenly spaced. The Fourier transform reconstruction in Figure 7b generates output truth table elements  $A$  given input truth table elements  $C$ . Using standard models of the holographic process it may be shown that the  $L \times P$  output truth table matrix  $A$  is related to the  $N \times P$  input truth table matrix  $C$  by

$$A = OR^*C \quad (2)$$

where  $O$  and  $R$  are  $L \times M$  and  $N \times M$  matrices describing the complex amplitudes used in recording the  $M$ -fold exposed hologram,  $m = 1, 2, \dots, M$ ,  $p = 1, 2, \dots, P$ , and  $+$  is the conjugate transpose operation. An important aspect of Eq. (2) is that although many exposures may be used to record the hologram, the ability of the hologram to represent input-output relationships is described by no more than the  $NL$  complex elements of  $OR^*$ . In the 2-bit multiplier, for example, where  $N = L = 4$  and  $P = 16$ , only 16 complex parameters are available to relate 64 input bits to 64 output bits. This suggests that not all possible truth tables are realizable in an optically recorded hologram of the type considered here. An analogous situation is that not all logic functions can be implemented by single threshold logic elements.<sup>6</sup>

It would be useful to at least approximately solve Eq. (2) for  $OR^*$  in terms of  $C$  and  $A$ . This matrix equation is generally over determined, and least-squares or pseudoinverse methods might be used to obtain an approximate solution. The (row by row) least-squares solution, for example, is

$$OR^* = AC^+(CC^+)^{-1} \quad (3)$$

While this solution may not yield the desired truth table realization in a lumped threshold system, it may serve as a starting point for a steepest descent or other computer search for desired solutions. Such solutions should maintain the desired input-output relationship when the matrix elements are varied over an acceptable tolerance range. The geometrical optics phase-only solution for the lumped threshold 2-bit multiplier described in Figure 3b is such a solution and may be used to derive an  $OR^*$  matrix in which all elements have unit magnitude:

$$OR^* = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & (-1 + \sqrt{3}i)/2 & 1 & (-1 - \sqrt{3}i)/2 \\ (-\sqrt{15} + i)/4 & (\sqrt{15} + i)/4 & (\sqrt{15} + i)/4 & (-\sqrt{15} + i)/4 \\ 1 & 1 & -1 & 1 \end{bmatrix} \quad (4)$$

A particular implementation of this solution for holographic recording is  $O =$  the  $4 \times 4$  identity matrix and  $R = (OR^*)^+$ . Note that although the above analysis implies three-dimensional holographic systems, integrated optical assemblies of diffracting elements similar in function to bulk holograms may be practical. This possibility is related to the observation that the multiple truth table "images" to be recorded and reconstructed, although often highly cross-correlated, may be relatively simple or low-resolution bright-spot-dark-spot patterns.

#### General Lumped Threshold Module Synthesis

Optical or computer generated hologram synthesis of the weighting or interconnecting units required for lumped threshold computation modules will generally require knowledge of the amplitude and phase patterns on the hologram that yield the correct truth table input-output behavior with maximum weight and threshold tolerances. In the case of the geometrical optics two-bit multiplier design of Figure 3, expressions governing input-output behavior were easily obtained. This favorable situation may be uncommon in the design of the generally smaller, more efficient, etc., lumped threshold modules for which geometrical optics approximations do not apply.

Consider, for example, the derivation of eight far-field holograms, each with the same two design parameters, that implement, in a lumped threshold system, the eight positive-threshold two-Boolean variable functions (i.e., the eight out of the sixteen functions for which two zero inputs yield a zero output). Figure 8 shows a simple format consisting of a screen with two pinholes separated by a distance  $y$ . One pinhole is covered by a phase-shifting film  $\theta$ ; there is a detector  $d$  and lower and upper mutually coherent point sources  $l$  and  $u$ . In the far-field approximation the distances  $b$  and  $y$  and the wavelength  $\lambda = 2\pi/k$  must be small compared to the distance  $s$ . With this approximation and with  $b$  fixed, the problem reduces to finding values of  $y$  and  $\theta$  such that the detected signals  $I_i$  for only source  $l$  on,  $I_u$  for only source  $u$  on, and  $I_b$  for both sources on have all six possible

inequality relationships. Referring to Figure 8 for definitions, the following approximate expressions may be derived:

$$\begin{aligned}
 A_I &= \exp i[k(w + s)] + \exp i[k(v + r) + \theta] \\
 A_U &= \exp i[k(w + s)] + \exp i[k(u + r) + \theta] \\
 I_I &= |A_I|^2 \approx (ks)^2(x^2 + ax)^2 + 2\eta(ks)(x^2 + ax) + \eta^2 \quad (5) \\
 I_U &= |A_U|^2 \approx (ks)^2(x^2 - ax)^2 + 2\eta(ks)(x^2 - ax) + \eta^2 \\
 I_B &= |A_I + A_U|^2 \approx 2(ks)^2 [(x^2 + ax)^2 + (x^2 - ax)^2] \\
 &\quad + 8\eta(ks)x^2 + 4\eta^2 - 4(ks)^2(ax)^2
 \end{aligned}$$

where  $x = y/s$ ,  $a = b/s$ , and  $\eta = \theta - \pi \approx 0$ . Figure 9 is a graph of the approximate expressions for  $I_I$ ,  $I_U$ , and  $I_B$  versus  $x$  for  $\lambda = 628$  nm,  $b = 10$   $\mu$ m,  $s = 10$  cm, and  $\eta = .004$ . Note that four of the six inequality relationships can be satisfied using the plotted values; the other two relationships can be satisfied for other values of  $\eta$ .

The example of Figure 8 and Eqs. (5) indicates the possible complexity of general (physical optics) lumped threshold module synthesis. Greater complexity may be anticipated if Fresnel rather than Fraunhofer diffraction conditions are allowed and if the input-output truth tables are large. One approach to such synthesis problems is to perform additional post-photodetection processing and to employ logical reduction and residue arithmetic techniques to reduce the effective size of the truth tables to be realized.<sup>7</sup> The approach considered here seeks alternatives to requirements for conversion into and out of residue arithmetic and for additional all-electronic processing.

Further work on lumped threshold logic should emphasize studies of the types and sizes of realizable truth tables and should seek general methods for synthesizing holograms such that the required truth tables are realized with acceptable threshold and other tolerances. A straightforward but perhaps limited approach to these studies is to investigate the number of holograms with specified resolution and cross correlation characteristics that can be multiplexed on a single recording medium. A more general approach may be to obtain expressions - generally large sets of overdetermined simultaneous nonlinear inequalities - that fully describe a desired optically implemented lumped threshold module and to find optimal solutions for them using nonlinear programming techniques. This approach will probably require the use of supercomputer facilities, but in many cases it may be the best approach for obtaining designs with optimum performance characteristics.

#### References

1. R. Arrathoon and M. H. Hassoun, "Optical Threshold Elements for Digital Computation," Opt. Lett. **10**: 143-145 (1984).
2. S. C. Gustafson, D. L. Flannery, and R. O. Winder, "Analysis of Threshold Logic for Application to Optical Computing," Proc. SPIE **564**, no. 24, (1985).
3. J. W. Goodman, F. J. Loonberger, S. Y. Kung, and R. A. Athale, "Optical Interconnections for VLSI Systems," Proc. IEEE **72**: 850-866 (1984).
4. D. L. Flannery, S. C. Gustafson, and L. M. Vail, "The Use of Multiple Associative Memories for Symbolic Processing," White paper, 15 pp., (9 Aug. 1985).
5. C. M. Verber, "Integrated Optical Approaches to Numerical Optical Processing," Proc. IEEE **72**: 942-953 (1984).
6. P. M. Lewis and C. L. Coates, Threshold Logic, Wiley, NY, pp. 458-467, (1967).
7. T. K. Gaylord, M. M. Mirsalehi, and C. C. Guest, "Optical Digital Truth Table Look-up Processing," Opt. Eng. **24**: 48-58 (1985).



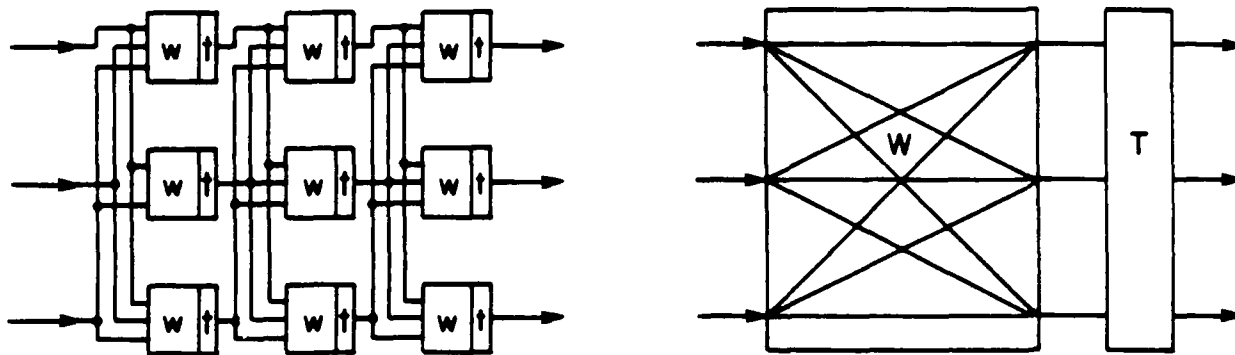


Figure 1. (a) Distributed threshold logic, (b) lumped threshold logic.

$x_1$	$x_0$	$y_1$	$y_0$	$z_3$	$z_2$	$z_1$	$z_0$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Figure 2. 2-bit multiplier truth table. The 12th row of the input and of the  $z_2$  column are boxed.

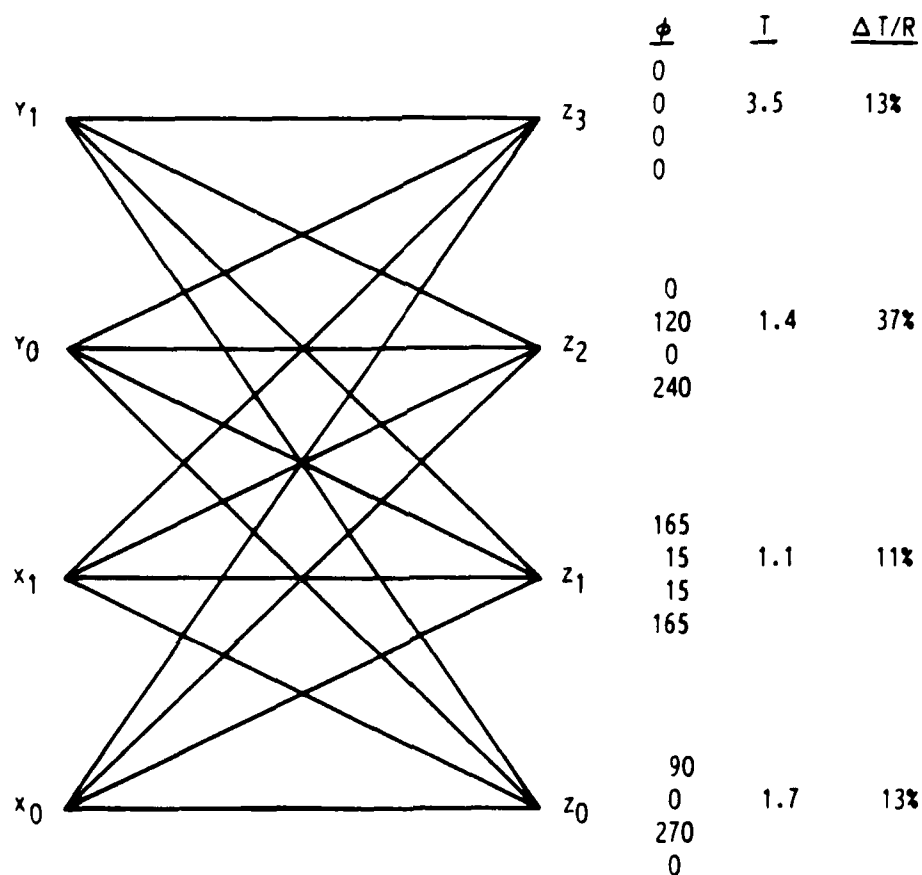


Figure 3. Lumped threshold 2-bit multiplier. (a) Interconnections from sources  $x$  to detectors  $z$ , (b) no-attenuation solution for interconnection phases  $\phi$ , detection thresholds  $T$ , and threshold tolerances  $\Delta T/R$ .

AD-A169 882

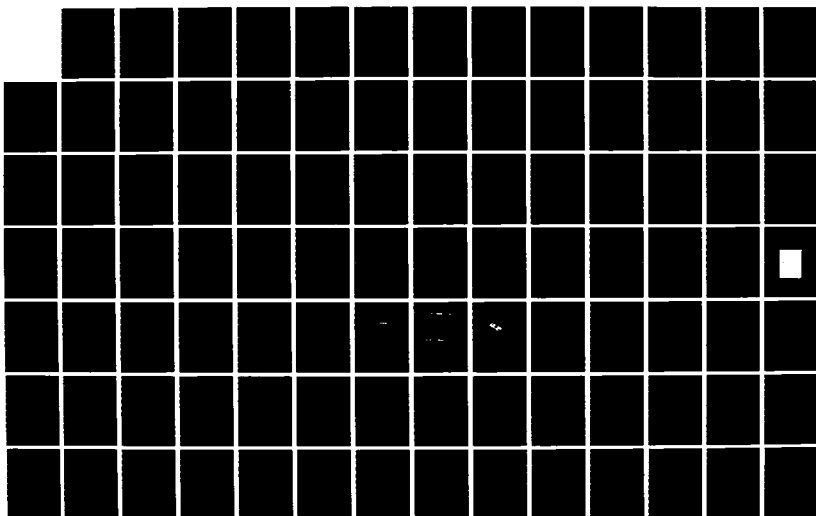
OPTICAL PROCESSING(U) DAYTON UNIV OH RESEARCH INST  
S C GUSTAFSON 31 DEC 85 UDR-TR-85-148 N00014-85-K-0479

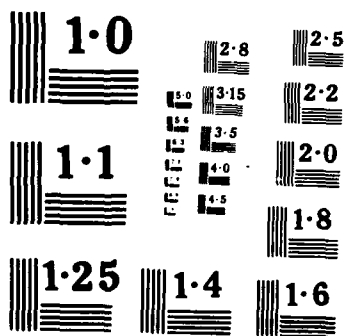
4/6

UNCLASSIFIED

F/G 9/2

NL





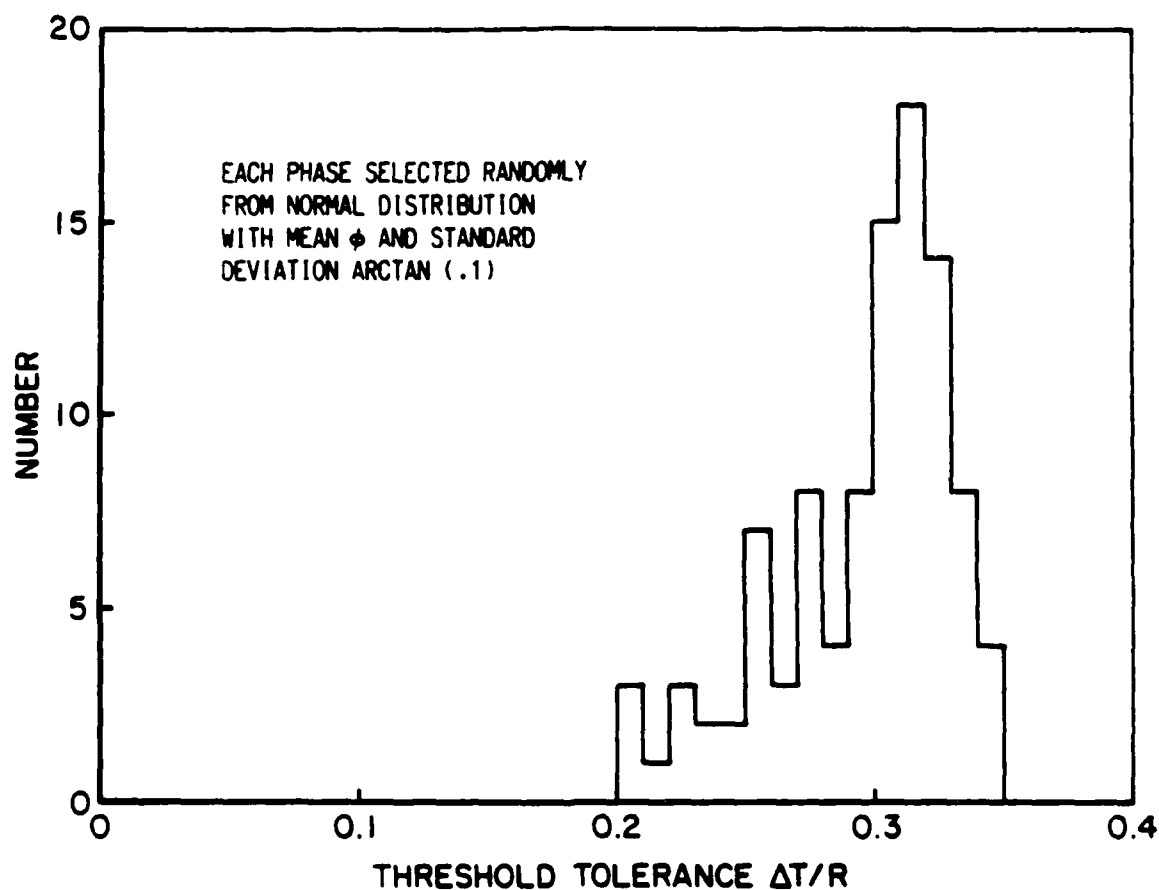


Figure 4. Histogram of threshold tolerance for output  $z_2$ .

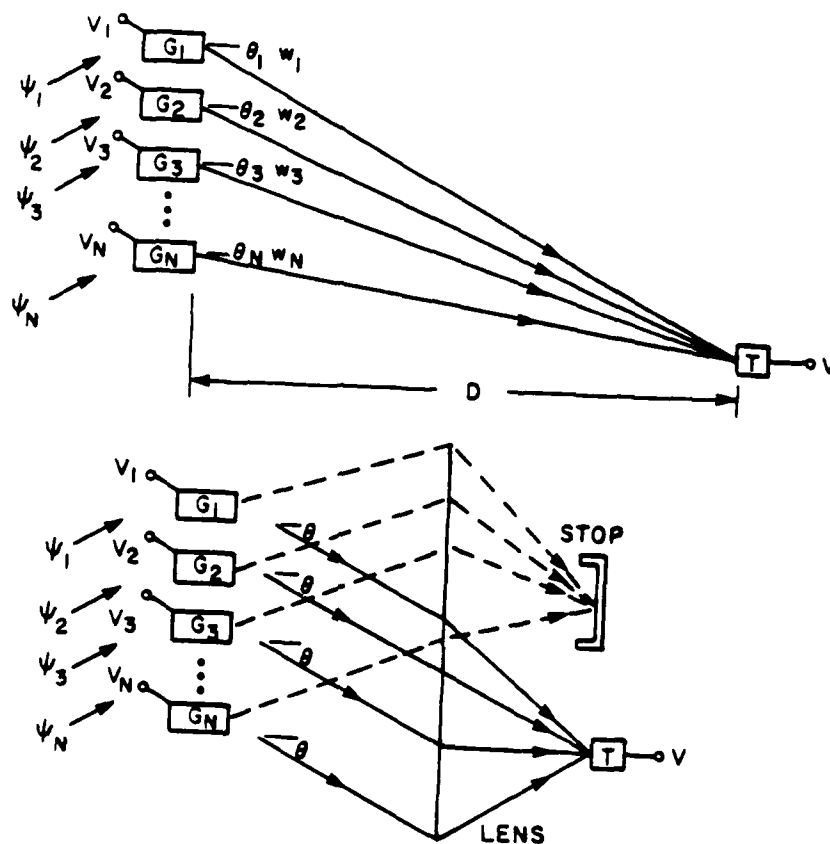


Figure 5. Integrated electro-optic grating arrays for programmable threshold logic. (a) Direct implementation, (b) arrays with lens.

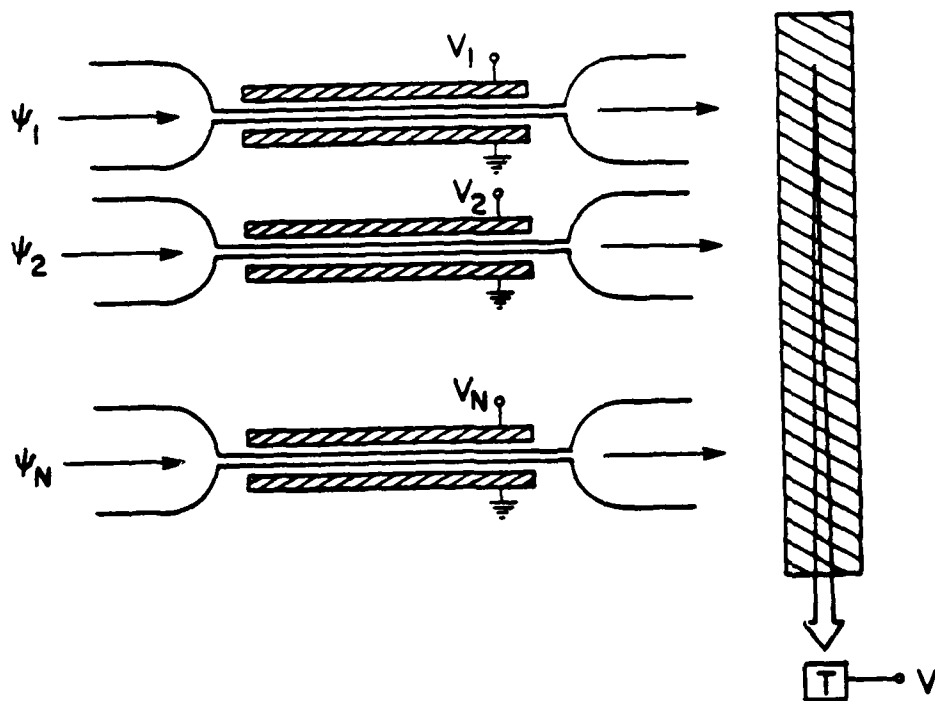


Figure 6. Integrated electro-optic channel arrays for programmable threshold logic.

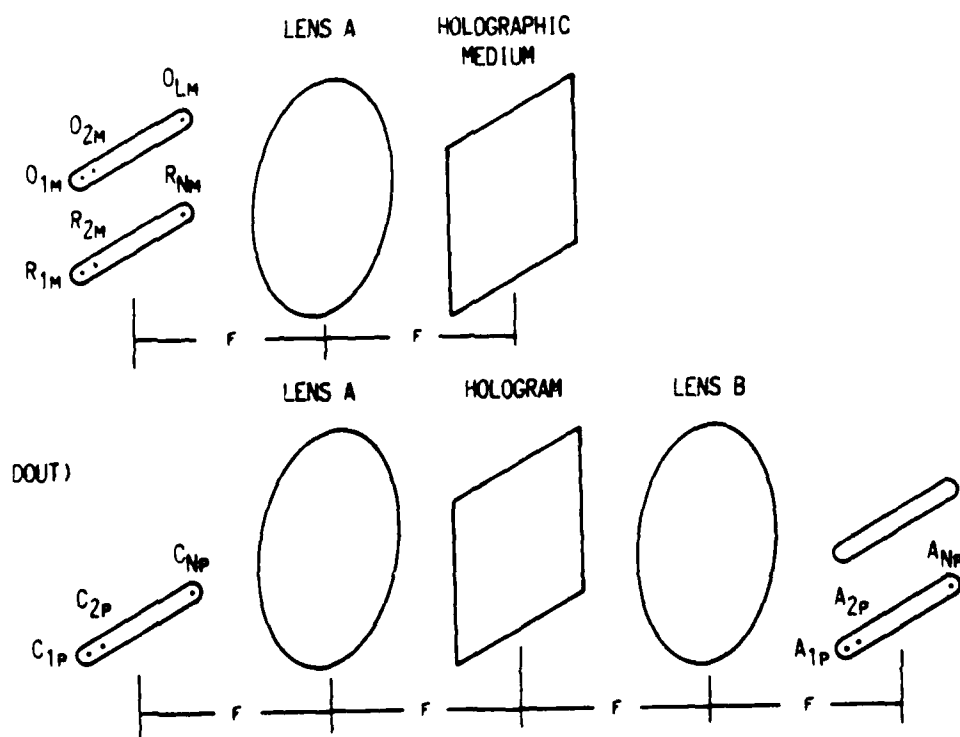


Figure 7. Holographic synthesis of lumped threshold logic. (a) Recording, (b) reconstruction (truth table readout).



123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100



123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100



**HOLOGRAPHIC WEIGHTING AND  
PHASE CONJUGATION FOR EXTERNAL  
THRESHOLDING ARCHITECTURES**

**Gordon R. Little  
University of Dayton  
Research Institute  
Dayton, Ohio 45469**

**Abstract**

Possible holographic implementations of the weighting operations required for external thresholding architectures are reviewed and extended, particularly for the coherent source, complex weight case. A four-input-bit experimental effort is described for this case that includes an improved phase stabilization/control scheme. The possible use of optical phase conjugation for significantly improving the performance of certain external thresholding architectures is also discussed.

## Holographic Weighting for External Thresholding Architectures

A previous investigation has considered the use of holographic interconnects to implement the weighting operations in threshold logic.<sup>1</sup> This work included an analysis of holographic weighting for the coherent source, complex weight case. A review and extension of this work is considered below.

A threshold logic element is taken to have  $N$  binary input variables  $\{a_n\}$   $n=1, N$  and a single binary output variable  $d$  given by

$$d = T\left(\sum_n w_n a_n\right) , \quad (1)$$

where  $T$  is a thresholding operation and  $w_n$  is the set of weights which are applied to the input variables. Defining  $c$  as the weighted sum of the input variables,

$$c = \sum_n w_n a_n , \quad (2)$$

and assuming real weights, the output variable can be written

$$d = u(c - t) , \quad (3)$$

where  $u$  is the Heaviside step function

$$u(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} , \quad (4)$$

and  $t$  is the threshold. Note that Equation (2) describes the inner product between a weight vector  $w$  and an input vector  $a$ .

In a coherent optical implementation, the weights and the weighted sum will, in general, be complex. In this case, a

number of thresholding schemes are possible, depending on how one chooses to partition the output space (the complex plane). The present investigation adopts a thresholding scheme based on the modulus squared of the weighted sum,

$$d = u(|c|^2 - t) \quad . \quad (5)$$

This scheme is appropriate if thresholding is accomplished electronically following standard photodetection.

In general, there will be many different input vectors which may be presented to the logic element, each with a specific desired output. In addition, we may wish to implement several logic functions in parallel. Denoting the number of input vectors by  $P$ , and the number of logic functions by  $M$ , we can express the weighted sum for the composite system by the matrix equation

$$C = WA \quad (6)$$

Here  $A$  is an  $N \times P$  matrix whose columns represent the  $P$  input vectors,  $W$  is an  $M \times N$  matrix whose rows represent the weight vectors for the  $M$  logic functions, and  $C$  is an  $M \times P$  matrix whose columns represent the  $M$  outputs for each of the  $P$  input vectors. Note that if all possible input vectors are permitted, then  $P = 2^N$ .

A fundamental step in the design of a threshold logic element is the determination of the weight and threshold values which yield the desired output. For each of the  $M$  logic functions there are  $P$  inequalities in  $N+1$  unknowns which must be solved. If the weights are real valued, the inequalities are linear, but if the weights are complex and if the thresholding scheme described by Equation 5 is used, then the inequalities are nonlinear. In either case, the determination of weights and thresholds which provide acceptable tolerances is a non-trivial problem. For a fully populated input space (i.e.  $P = 2^N$ ), and for

$N \geq 3$  appropriate solutions for specific logic functions may not exist.<sup>2</sup>

A possible holographic implementation of the weighting in a threshold logic element is depicted in Figure 1b. Here, the binary inputs are a set of  $N$  mutually coherent point sources located in the front focal plane of the input lens and the weighted sums for the  $M$  outputs are represented by a set of  $M$  spots in the back focal plane of the output lens. Regarding the outputs as a generalized reconstruction of a set of object sources, a potential scheme for fabricating the weighting hologram would be to superimpose a number of exposures from an array of  $M$  object sources and  $N$  reference sources as shown in Figure 1a. It is this approach which has been analyzed.

To obtain equations describing the system the hologram is taken to be a thin, amplitude hologram and the Fourier transform geometry shown in Figure 3b is assumed. Taking  $R_{ni}$  and  $O_{mi}$  to represent the complex amplitudes used in the  $i$ th exposure of the  $n$ th reference source and the  $m$ th object source, the transmittance  $T$  of the hologram can be written

$$T = \alpha \sum_i \left| \sum_n R_{ni} e^{ik_n \cdot r} + \sum_m O_{mi} e^{ik_m \cdot r} \right|^2 \quad (7)$$

where  $k_n$  and  $k_m$  are wave vectors of the reference and object beams incident to the holographic recording medium,  $r$  is the position vector in the medium, and  $\alpha$  is an unimportant constant. If the reconstruction sources are positioned at the reference source locations, the complex optical amplitude transmitted by the hologram,  $\psi(r)$ , is

$$\psi(r) = \left( \sum_n A_{np} e^{ik_n \cdot r} \right) T \quad (8)$$

or

$$\begin{aligned}
 \psi(r) = & \sum_n \sum_{n'} \sum_{n''} \left( \sum_f R_{n'f} R_{n''f}^* \right) A_{np} e^{i(k_n + k_{n'} - k_{n''}) \cdot r} \\
 & + \sum_n \sum_{n'} \sum_m \left( \sum_f R_{n'f} O_{mf}^* \right) A_{np} e^{i(k_n + k_{n'} - k_m) \cdot r} \\
 & + \sum_n \sum_m \sum_{n''} \left( \sum_f O_{mf} R_{n''f}^* \right) A_{np} e^{i(k_n + k_m - k_{n''}) \cdot r} \\
 & + \sum_n \sum_m \sum_{m'} \left( \sum_f O_{mf} O_{m'f}^* \right) A_{np} e^{i(k_n + k_m - k_{m'}) \cdot r}
 \end{aligned} \tag{9}$$

Of the  $N(N+M)^2$  plane wave terms in Equation (9) only the M waves in the third line for which  $n' = n$  are of interest. It is these waves which form the reconstructed images of the object points. It is seen that these terms can be written

$$\sum_m \left( \sum_n \sum_f O_{mf} R_{nf}^* A_{np} \right) e^{i k_m \cdot r} \tag{10}$$

The amplitudes of the M plane waves described by Equation (10) are the amplitudes of the reconstructed objects. It is seen that the action of the multiple exposure hologram can be expressed in a matrix equation

$$C = OR^\dagger A, \tag{11}$$

where  $\dagger$  is the Hermitian transpose operation and O and R are matrices whose columns represent the complex amplitudes of the object and reference sources used in each exposure. Comparing Equations 6 and 11 it is seen that the weight matrix and the hologram matrices are related by  $W = OR^\dagger$ . It should be noted that the use of thick phase holograms along with careful source geometry selection will reduce the strength of the undesired

terms to low levels; the output wave vectors for these terms will not satisfy the Bragg conditions.

From this analysis, it is seen that if the sets of weights which realize specific sets of threshold logic functions are known, then a holographic implementation of the weighting can be designed. In fact, there are many possible designs, depending on how the weight matrix is factored. A factorization which emphasizes the independence of the  $M$  logic functions and which imposes minimal phase control constraints in fabrication is to let  $O$  be the  $M \times M$  identity matrix and let  $R^\dagger$  be the  $M \times N$  weight matrix. Then there would be  $M$  exposures, each writing a separate logic function onto the hologram. Since the number of elements in the weight matrix is limited, a reasonably low number of hologram exposures is required to implement interesting, realizable threshold logic functions. A full precision 16-bit multiplier, if realizable, would require only 32 exposures. The content addressable memory approach,<sup>3</sup> on the other hand, requires  $\sim 10^{10}$  exposures if standard arithmetic is used and  $\sim 10^3$  exposures if logical reduction techniques and residue arithmetic are used. Critical questions which remain, however, include the realizability of such high bit-number arithmetic operations and the tolerance of such a scheme to phase and amplitude errors in fabrication and readout. It should be noted that there are many interesting problems in such areas as target recognition and tracking where the input space is limited ( $P \ll 2^N$ ), making realizability of a given threshold logic function much more likely.

#### Experimental Effort on External Thresholding Architectures

To investigate the concepts derived from the analysis and to explore the problems involved in reducing the holographic weighting scheme to practice, a limited experimental effort was carried out. This effort was restricted to threshold logic functions of four input bits and was specifically focused on the

four functions representing the output bits from a two-bit multiplier. All of these functions are realizable using complex weights but only three are realizable with real weights.

Clearly, the relative phases of the sources used in fabricating the weighting hologram must be controlled and the precision of control is dependent on the tolerance requirements in the thresholding operation. Preliminary experiments also indicated that thermally induced phase drifts in the laboratory were sufficiently large to present significant difficulty in carrying out the experiments. Accordingly, an active phase stabilization/control scheme was devised. The scheme described by MacQuigg,<sup>4</sup> was modified to improve the phase control precision from  $\sim 10^\circ$  to  $\sim 1^\circ$ . The phase stabilization/control system as applied to two sources is depicted in Figure 2. A phase control grating is fabricated using the two sources and is rotated to produce a fringe pattern on a detector. The phase of one of the sources is dithered at the reference oscillator frequency of a lock-in amplifier to produce a small motion of the fringes. The resultant oscillatory detector output is fed into the lock-in amplifier and the lock-in amplifier output serves as the error signal to stabilize the DC phase of the dithered source relative to the other source. Phase control is achieved by translating the detector within the fringe pattern. Methods based on this scheme<sup>5</sup> may ultimately be important in high-speed optical computing (perhaps in integrated optical implementations) where mutually coherent sources are used.

The layout that has been constructed for the 2-bit multiplier experiment is shown in Figure 3. A prism arrangement is used to generate the eight sources required to construct the weighting hologram, and piezo-electric driven mirrors are used to implement the phase stabilization/control. Using the hologram design where  $R^\dagger$  is the weight matrix, only three control loops for the reference sources are needed.

The primary goal of the present experimental effort was to demonstrate that optically fabricated holograms could be used for complex valued weighting in simple threshold logic modules. One question raised in the course of the effort is whether processing induced distortions in the weighting hologram will significantly degrade performance. If so, the number of potentially useful hologram fabrication techniques may be limited. Other questions which remain to be investigated include the degree of threshold tolerance which is achievable and the precision requirements for phase and amplitude control of the sources during fabrication and read out.

An analysis of the realizability of logic functions within the holographic weighting context and an examination of the impact of realizability on architectures are fundamentally important. Topics include possible improvements in realizability using nonlinear hologram recording and alternative thresholding techniques for coherent weighting. The set of logic functions involved in multiplying two 4-bit numbers may be viewed as a functional module, and combinatorial logic architectures which use such modules to perform higher order operations merit further study.

The holographic weighting schemes investigated thus far have involved spreading each weight element over the entire hologram. Improved performance may be realized if the weight elements are separated. An analysis of this approach would include treatment of storage density questions, examination of computer generated holograms for coherent and incoherent weighting schemes, and design of simple architectures.

Most of the analytical and experimental work performed to date has concentrated on thin amplitude holograms based, for example, on silver halide plate technology. However, achievement of the storage densities necessary for practical optical computing will probably require the use of thick phase holograms and



photorefractive material technology. Such holograms have the advantage that no processing (i.e., developing) is required, and thus processing-induced distortions would be nonexistent. It is therefore important that the special features, capabilities, and constraints implied by the use of photo-refractive materials in holographic weighting be considered. Such consideration would include the analysis of thick phase holograms for coherent weighting and the design of architectures to utilize multiple passes through the hologram for implementing selected logic functions.

#### Phase Conjugation for External Thresholding Architectures

Optical phase conjugation (OPC) is an important emerging technology which may find wide application in optical computing and signal processing. Presently identified OPC capabilities of interest include phase distortion correction, lensless imaging, gain, and use in associative memories.<sup>6,7</sup> It is possible that eventual implementations of the popular circulating packet architectures may include phase conjugate devices for level restoration (gain), for module interconnection and input/output (lensless imaging), and for prefiltering (associative memories).<sup>8</sup>

Of immediate interest is the recent demonstration by Dunning et al.<sup>9</sup> of an all-optical associative memory. In this scheme, depicted in Figure 4,<sup>10</sup> a holographic memory is placed within a resonant cavity bounded by phase conjugate mirrors. A pattern is introduced to the holographic memory element using a beam splitter, and several reconstructed reference beams are produced by the hologram, depending on the degree of match between the input pattern and the stored patterns. A lens images the hologram onto one of the phase conjugate mirrors, which has a threshold level for conjugation. The threshold level is set so that only the strongest reconstructed reference beam resonates in the cavity, and the system output is the stored pattern which best matches the input.

Two aspects of this scheme appear to be directly applicable to external thresholding architectures. First, note that if the resonator output is taken from the other side of the holographic memory, the system becomes a look-up device, i.e., a content-addressable memory whose performance may be significantly improved by the thresholding action of the resonator cavity. It may be possible, for example, to store many more patterns within the memory than would be possible without the resonator. Second, note that the resonator could be used to form a thresholding device. In this case, the hologram and lens could be eliminated from the system and the input could be a spatially modulated beam of light. In areas where the input intensity is above threshold, the system would resonate and the output would be high, but in areas where the input intensity is low, the system would not resonate and the output would be low. Depending on the characteristics of the thresholding phase conjugate mirror, significant improvement over other thresholding schemes may result.

## REFERENCES

1. S. C. Gustafson, J. A. Kirk, G. R. Little, R. P. Kenan, C. M. Verber, "Optical Implementations of Lumped Threshold Logic," Proc. SPIE 564, pp. 157-166, San Diego, CA, August 1985.
2. S. C. Gustafson, D. L. Flannery, G. R. Little, S. C. Cartwright, L. M. Vail, J. S. Loomis, "Optical Processing Research at University of Dayton Research Institute," ONR/SDI Optical Processing Meeting, Washington, DC, 14 October 1985.
3. M. M. Mirsalehi, T. K. Gaylord, "Truth-Table Look-up Parallel Data Processing Using an Optical Content-Addressable Memory," to be published, App. Opt., March 1986.
4. D. R. MacQuigg, "Hologram Fringe Stabilization Method," Appl. Opt., 16, pp. 291-292, February 1977.
5. G. R. Little, "Phase Stabilization and Control Technique with Improved Precision," submitted to Appl. Opt., December 1985.
6. D. M. Pepper, "Nonlinear Optical Phase Conjugation," Laser Handbook, 4, M. L. Stitch, M. Bass, eds., North Holland, 1985.
7. B. Ya Zel'dovich, N. F. Pilipetsky, V. V. Shkunov, Principles of Phase Conjugation, Springer-Verlag, 1985.
8. D. L. Flannery, L. M. Vail, "All Optical Decision Elements and Optical Computing," in preparation.
9. G. J. Dunning, E. Marom, Y. Owechko, B. Soffer, "All-Optical Associative Holographic Memory with Feedback Using Phase Conjugate Mirrors," WT8, Opt. Soc. Ann. Meeting, Washington, DC, 16 October 1985.
10. D. M. Pepper, "Applications of Optical Phase Conjugation," Sci. Am., 254, pp. 24-83, January 1986.

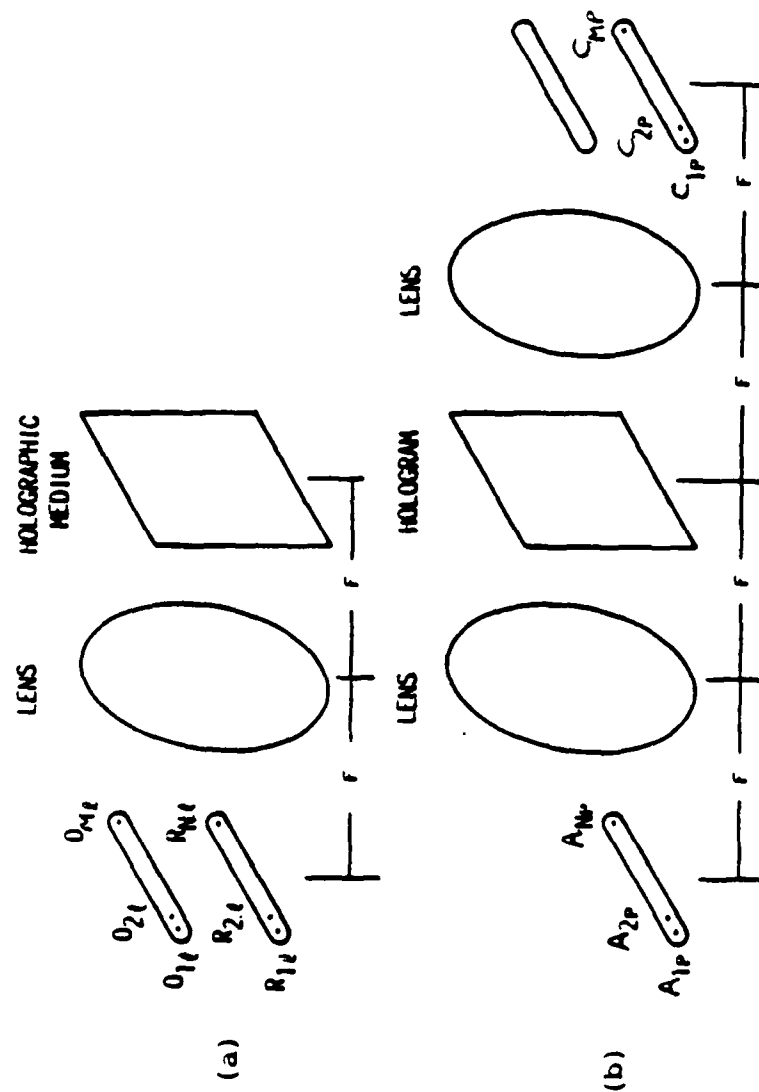


Figure 1. Holographic implementation of weighting in threshold logic: (a) recording, (b) reconstructing.

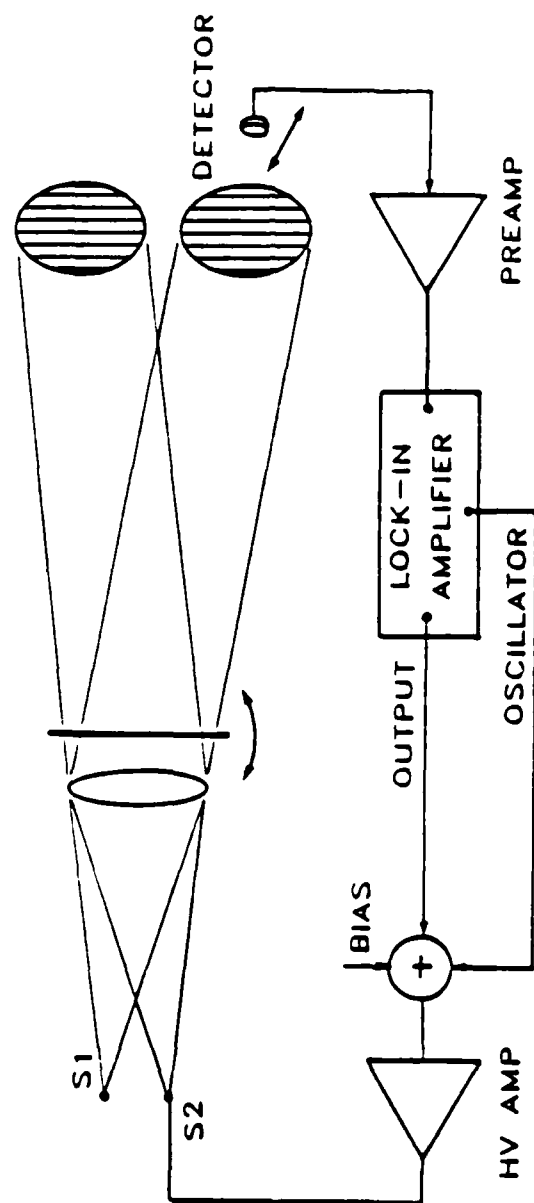


Figure 2. Phase stabilization/control scheme.

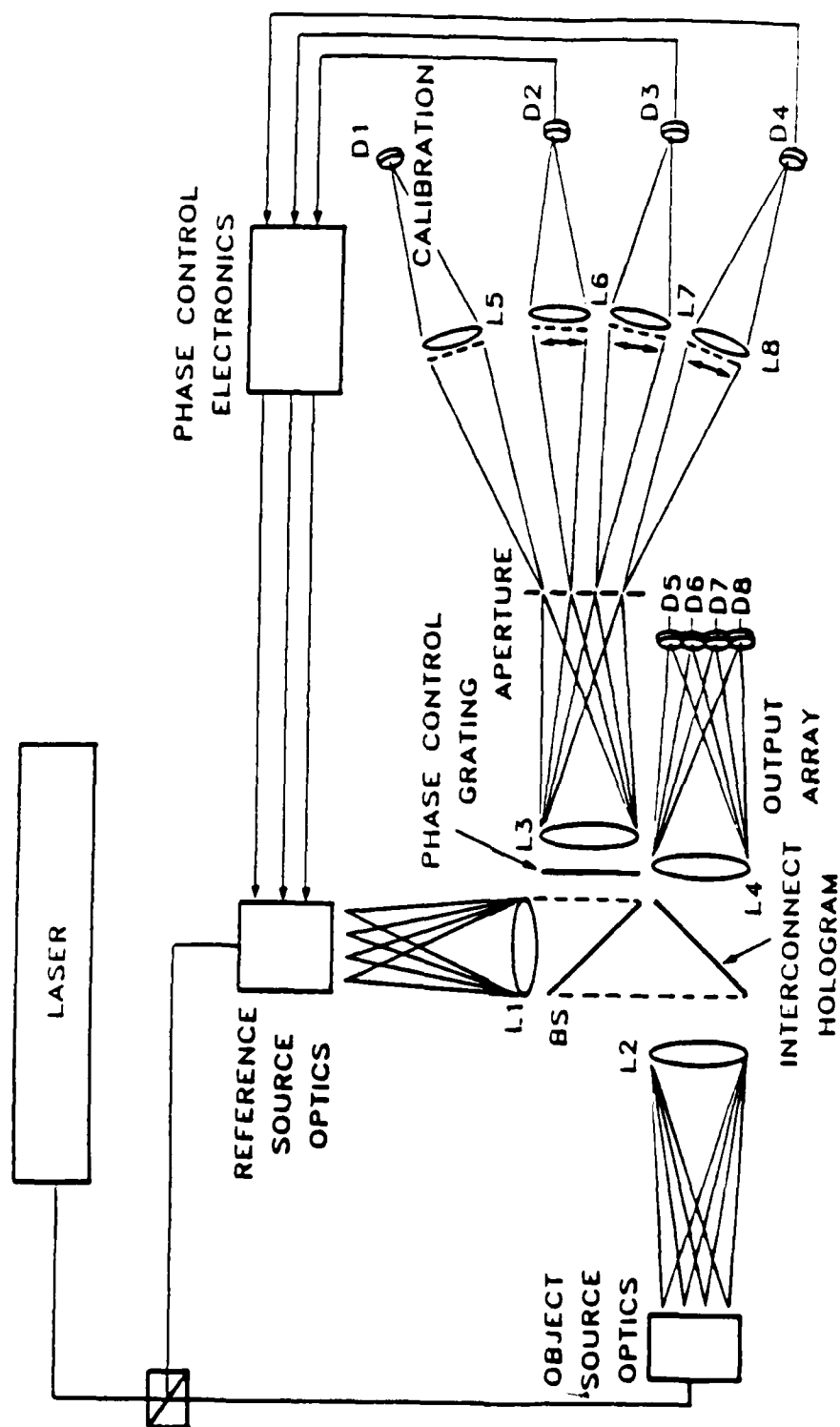
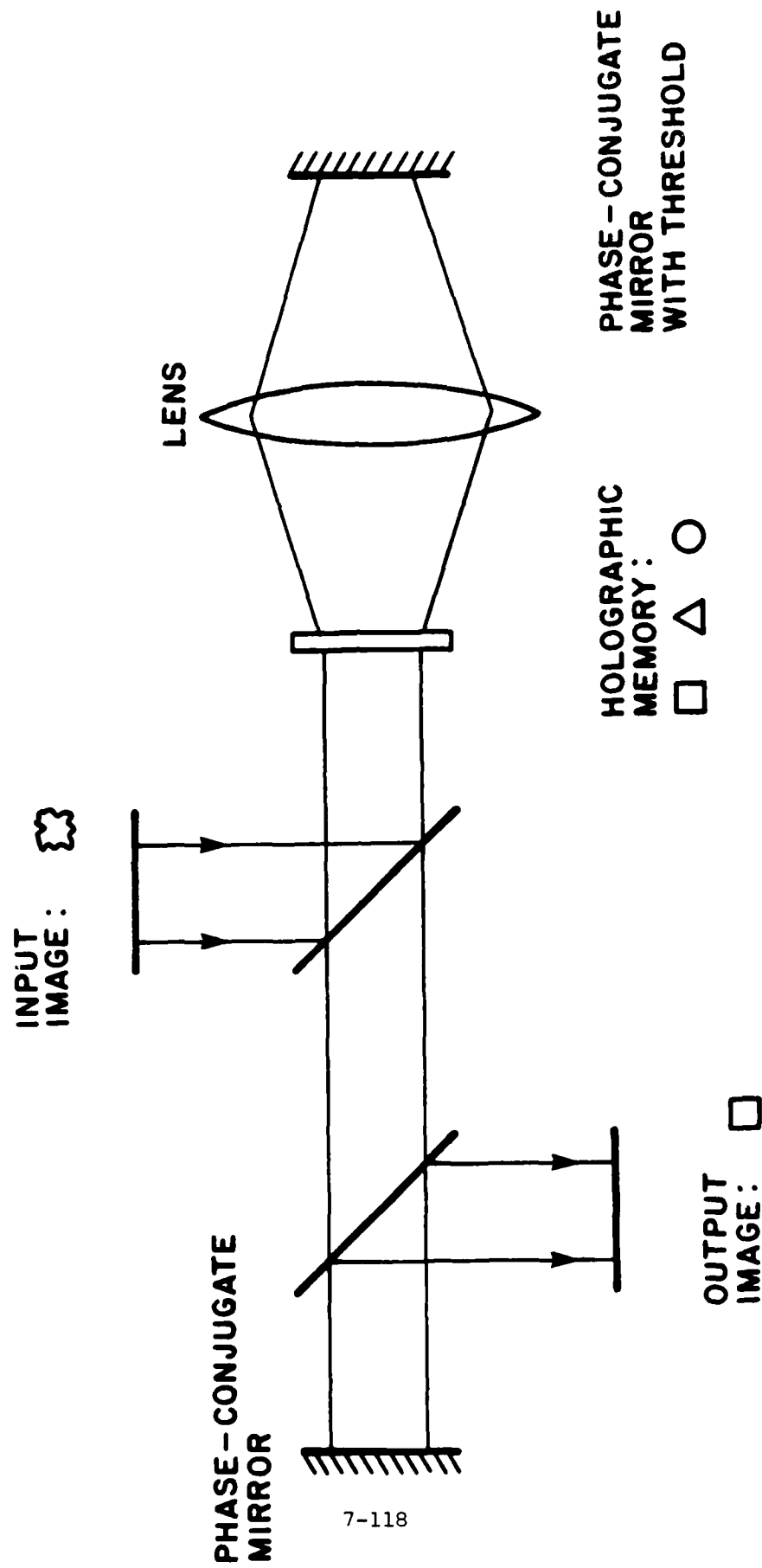


Figure 3. Layout of two-bit multiplier optical breadboard.



7-118

Figure 4. All-optical associative memory using phase conjugation.<sup>10</sup>

ROBUST TRACKING USING  
ELECTRO-OPTICALLY IMPLEMENTED NEURAL NETWORKS

L. Maugh Vail and David L. Flannery  
University of Dayton  
Research Institute  
Dayton, Ohio 45469

Abstract

The potential of electro-optically implemented Grossberg neural networks is assessed. These networks can perform adaptive pattern recognition using both short-and long-term memory: long-term memory stores feature vectors corresponding to many target patterns; short-term memory adaptively tracks the evolution of these feature vectors. The possible design, fabrication, and testing of an experimental neural network simulator system which contains an electro-optic linear algebra processor as a major component is also assessed. Compared to current processors based on Kalman filtering or related techniques, this type network should form the basis for an adaptive multi-sensor processor having a high order of "intelligence" for pattern recognition and tracking tasks.



## Introduction

The neural network models suggested by Grossberg, [1976a, 1976b, 1980] have potential to perform complex pattern recognition and tracking tasks required by strategic defense surveillance, acquisition, and tracking scenarios. These networks will have generic applicability to many problem areas involving similar mathematical structures. For the SDI a likely configuration would involve output feature values from several sensor preprocessing systems forming the input vector to the network. The network type assessed in this paper can perform adaptive pattern recognition and has both short- and long-term memory, which distinguishes it from other notable models [Hopfield, 1982; Willshaw, 1981; Kohonen, 1983]. The long-term memory (LTM) is of the correlation type [Grossberg, 1976a, 1976b, 1981; Kohonen, 1981, 1983] and can store many reference patterns representing feature vectors corresponding to specific threat or target patterns. The short-term memory (STM) is adaptive and can track the evolution of feature vectors.

A key aspect of assessment is the implementation of a high-level control structure in the network. Such a structure would constrain the adaptation range of the lower levels to prevent tracking of patterns beyond application-dependent acceptable bounds. This combination of features comprises the basis for an adaptive multisensor control and processing system which should have an "intelligence" of a higher order than that obtained with current techniques such as Kalman filtering. The use of optical algebraic processing as a major part of an experimental neural simulator which could be exercised to study network behavior is also assessed. The remaining sections of this paper thus present the mathematical basis of the model and describe a possible hybrid optical/digital experimental system.

### Mathematical Description

At some instant  $t$  we are given an  $N \times 1$  input vector  $\underline{u}(t)$  and a set of  $M$  normalized  $N \times 1$  pattern vectors  $\{p_0(t), p_1(t), \dots, p_{M-1}(t)\}$ . We wish to know whether  $\underline{u}(t)$  contains one of the patterns  $p_i(t)$  and if so which one. Since the patterns themselves may be slowly time-varying, we wish to have our system reflect these slow changes as well as track the input vector  $\underline{u}(t)$ . Let  $\underline{u}(t)$  have the structure  $[u_0(t), u_1(t), \dots, u_{N-1}(t)]^T$ . Because the individual components  $u_i(t)$  and  $u_j(t)$  are possibly correlated in ways that do not represent any of the desired patterns  $\{p_m(t)\}$  we wish to construct a state vector  $\underline{x}(t) = [x_0(t), \dots, x_{N-1}(t)]^T$  whose components  $x_i(t)$  and  $x_j(t)$ ,  $i, j = 0 \dots N-1$ , are correlated in a way which emphasizes some one of the desired patterns. In other words, noisy, unwanted correlations between individual components of  $\underline{u}(t)$  are suppressed in the transformation  $T : \underline{u}(t) \rightarrow \underline{x}(t)$ . It is required that the state vector  $\underline{x}(t)$  be based on the input, and the system is to find patterns inherent in  $\underline{u}(t)$  and track them. Since  $\underline{x}(t)$  could lock onto a false pattern, i.e., one not contained in  $\{p_m(t)\}$  it is desired that  $\underline{x}(t)$  be monitored by the system for validation of its tracking activity. For example: The input signals may become distorted through momentary, unexpected interference that could produce moving ghost reflections. This may cause  $\underline{x}(t)$  to consist of aspects distributed across two or more patterns  $\{p_m(t)\}$  in such a way that a hybrid pattern  $q(t)$  is represented, and short-term memory may begin to track this hybrid. The monitoring function guards against this kind of instability by comparing the adapted short-term pattern estimate  $\underline{x}(t)$  against the set of known patterns  $\{p_m(t)\}$ .

A neural net based on the Grossberg model is suggested, to address this problem. The model is characterized by the equations

$$\dot{x}_i(t) = -\alpha x_i(t) + (\beta - x_i(t)) [f_i(x_i(t)) + u_i(t)] \quad (1)$$

$$\dot{M}_{ij}(t) = \begin{cases} -\nu M_{ij}(t) + \rho x_i(t)x_j(t) \\ \text{if } \|s_i(t) - p_m^{(i)}(t)\| < \epsilon \text{ for all } m \\ 0, \text{ otherwise} \end{cases} \quad (2)$$

$$s_i(t) = \sum_{j=0}^{N-1} M_{ij}(t) x_j(t) \quad (3)$$

for  $i, j = 0, \dots, N-1$  [Grossberg 1975, 1976a, 1977; Cohen, 1983] and as shown in Figure 1. These equations implement a short-term memory in the vector  $\underline{x}(t)$  and a long-term memory by means of a "connectivity matrix"  $\underline{M}$  [Ellias & Grossberg, 1975; Grossberg, 1976a and 1976b, 1980, 1981]. The short-term memory acquires its character through choice of the function  $f(w)$  and allows the options of contrast enhancement, pattern selection and matching, or noise suppression. A sigmoid function, e.g.,  $f(w) \propto [\arctan w]_{p.v.}$ , has been shown to combine these options over ranges of  $w$  where they are most needed, i.e., emphasizing contrasts when  $w$  is very weak, pattern selection and replication when  $w$  is of moderate intensity, and staying below some saturation level when spikes occur [Grossberg, 1975]. The short-term memory has relatively fast dynamics and tracks some pattern in the input, but it is blind to the identity of this pattern.

The matrix  $\underline{M}$  corresponds to the long-term memory of the

system. It acts as a monitor for the tracking part of the system and can also adapt to reflect slow changes in stationarity of the patterns  $\{p_m(t)\}$ . Initially  $\underline{M} = p_0(t)p_0(t)^T + p_1(t)p_1(t)^T + \dots + p_M(t)p_M(t)^T$  where  $M \ll N$ . Let us suppose for ease of argument that the  $\{p_i(t)\}$  are orthonormal, i.e.,  $p_i(t)^T p_j(t) = \delta_{ij}$  where  $\delta$  is the Kronecker delta. Then we see that  $\underline{M} p_m(t) = [p_0(t)p_0(t)^T + \dots + p_m(t)p_m(t)^T + \dots + p_M(t)p_M(t)^T] p_m(t) = p_m(t)$ . It has been shown that if a vector  $y(t)$  is not identical to any of the  $\{p_m(t)\}$  then the operation  $\underline{M} y(t)$  will produce a vector which lies "closest" to one of the  $p_m(t)$  [Kohonen 1981, 1983, Grossberg, 1976a]. The degree of acceptability of this closeness may be determined as follows: We form the output vector  $\underline{s}(t) = \underline{M} \underline{x}(t)$ . If the short-term memory vector  $\underline{x}(t)$  is tracking one of the desired patterns  $p_m(t)$ , i.e., if  $p_m(t)$  has been detected as present in  $\underline{u}(t)$  then  $\| \underline{s}(t) - p_m(t) \| < \epsilon$  for some  $\epsilon$  and some  $m$  selected from  $(0, \dots, N-1)$ , i.e.,  $\underline{s}(t)$  is "close" to  $p_m(t)$ . If this closeness requirement is met, we update the matrix  $\underline{M}$ . Otherwise we reset  $\underline{x}(t)$  and begin the search for another pattern.

#### Coupling of Slow and Fast Dynamics

The monitoring of the fast tracker, STM, by the long-term memory component, LTM, requires that the dynamics of both be coupled into a larger system. When the fast and slow parts of a system are tightly coupled, boundary layer instabilities arise such as are found in turbulence, shears, etc. To avoid these phenomena, tightly coupled systems are often accompanied by restrictions which allow only one part of the system to make a transition at a given instant in time. An example is a model which construes Equation (2) as a running time estimation of an underlying wide-sense stationary process. The model featured here avoids many of these restrictions by maintaining a "loose" coupling of the two time scales.

To clarify this point further, note that Equation (1) does not contain an explicit reference to LTM. It is worth noting

that classical approaches have frequently maintained a tight coupling of the two time scales, eg., in recursive identification techniques and in state variable feedback schemes.

### The Role of LTM and STM in Recursive Identification Systems

Recursive identification schemes typically take as their point of departure some variant of Wiener filtering theory or mean-square estimation. The signal model is that the input  $u_i(t)$  contains a signal  $p_i(t)$  and a noise term  $n_i(t)$ :  $u_i(t) = p_i(t) + n_i(t)$ . The input signal  $u_i(t)$  thus is presumed to contain a deterministic part  $p_i(t)$ , such that knowledge of neighboring values  $p_j(t)$  allows us to construct  $p_i(t)$ , for all  $j, i$ . However,  $u_i(t)$  also contains a part  $n_i(t)$  which randomly fluctuates and for which knowledge of its values at any and all other positions  $j$  and times  $t-\tau$  are of no help in estimating  $n_i(t)$ . Viewed separately,  $n_i(t)$  has instantaneously fast dynamics whereas  $p_i(t)$  has relatively slower dynamics. Thus,  $u_i(t)$  is some sort of mixed process whose deterministic part may be estimated by a parameterized process as follows:

$$\hat{p}_i(t) = \sum_{j \neq i} x_j(t) u_j(t)$$

where  $\underline{x}(t)$  is a parameter vector. If we know  $\underline{x}(t)$  then we can form the above estimate and at least approximately reconstruct the signal  $p_i(t)$ . Wiener filtering theory does this by requiring that  $\underline{x}(t)$  be chosen such that the mean squared error  $\epsilon_i = \|p_i(t) - \hat{p}_i(t)\|^2$  be minimized. The result of this minimization process is the familiar Wiener-Hopf equation (Papoulis, 1984)

$$b_i(t) = \sum_j R_{uu}(i, j) x_j(t)$$

where

$$b_i(t) = E(p_i(t)u_i(t)) \text{ and } R_{uu}(i, j) = E(u_i(t)u_j(t))$$

If  $\underline{u}(t) = \underline{p}(t) + \underline{n}(t)$  is wide sense stationary and if  $\underline{n}(t)$  is a spatially uncorrelated noise process then

$$E\{u_i(t)u_j(t)\} = R_{uu}(i,j) \approx \underline{p}\underline{p}^T = \underline{R}$$

for some stationary pattern  $\underline{p}$ . Hence, the vector equation  $\underline{b} = \underline{R}\underline{x}$  has the solution  $\underline{x} = \underline{R}^{-1}\underline{b}$ . To estimate this recursively a running time estimate  $\underline{R}(t) = \underline{R}(t-1) + \rho \underline{u}(t) \underline{u}(t)^T$  can be formed and then  $\underline{x}(t) = \underline{x}(t-1) + \mu \underline{R}^{-1}(t) \underline{b}(t)$  can be computed where  $\underline{b}(t)$  is either known a priori or estimated, e.g.,  $\underline{b}(t) = \underline{x}(t-1) \underline{x}(t-1)^T \underline{u}(t)$ . There are many variants of this general procedure. (Ljung, 1983; Monzingo and Miller, 1980). Note that  $\underline{R}$  plays the role of LTM and that its dynamics are by hypothesis confined to functional variations within a temporal ensemble. Although this model does not cleanly separate LTM ( $\underline{R}$ ) and STM ( $\underline{x}$ ) in terms of slow and fast dynamics, the rates of convergence to steady state of the  $x_i$  are known to be inversely proportional to the largest eigenvalue of some ideal  $\underline{R}$ . It is clear that  $\underline{x}$  cannot reach a steady state neighborhood if  $\|\underline{R}(t) - \underline{R}(t-1)\|$  does not become small for large  $t$ . This can only be guaranteed by placing assumptions on the model, e.g., the process is ergodic.

The analogue to this type of recursive identification procedure for the proposed neural net model is to replace Equation (1) with an equation of the form

$$\dot{x}_i(t) = -\alpha x_i(t) + x_i(t) \sum_{j=0}^{N-1} Q_{ij}(t) u_j(t) \quad (1a)$$

where  $\underline{Q} = \underline{M}^{-1}$  and Equation (2) with

$$\dot{M}_{ij}(t) = -\alpha M_{ij}(t) + \rho u_i(t) u_j(t) \quad (2a)$$

Note that whereas the recursive identification schemes use LTM to store one pattern, the neural net model can store a multiplicity

of patterns. In other words the parameter vector  $\underline{x}(t)$  in the recursive identification scheme plays the role of a retrieval key whereas in the neural net model LTM is now hetero-associative and initialized e.g., by  $\underline{M} = \underline{p}_0 \underline{x}_0^T + \dots$  instead of by  $\underline{M} = \underline{p}_0 \underline{p}_0^T + \dots$ .

In Equation (2) LTM is allowed to decay slowly or to be reinforced by patterns which prevail in STM, provided they are sufficiently close to one of the initial patterns ( $\underline{p}_i$ ). Some variants of Equation (2) are

$$\dot{M}_{ij}(t) = -\nu_j M_{ij}(t) + \rho s_i(t) x_j(t) \quad (2b)$$

and

$$\dot{M}_{ij}(t) = -\nu_j M_{ij}(t) + \rho s_i(t) s_j(t) \quad (2c)$$

which guarantee that LTM is reinforced by patterns which are projections onto some of the ( $\underline{p}_0$ ) since  $\underline{s}(t) = \underline{M}(t)\underline{x}(t)$ . On the other hand, we might want to allow LTM to slowly acquire unforeseen patterns. Thus an entirely new and unanticipated pattern will be recorded and monitored by the system provided that it is persistent enough. Such a situation would be reflected in

$$\dot{M}_{ij}(t) = -\nu M_{ij}(t) + \rho x_i(t) x_j(t) \quad (2d)$$

Finally, the neural net could make use of two LTM's, one which has a permanent set of patterns ( $\underline{p}_m$ ) and a second which is allowed to slowly acquire STM patterns. Such a system would have a permanent record of its initial patterns but would have the capability of responding to unforeseen situations. In addition the system could activate one set of "routine" control signals when one of the original patterns is recognized and activate a second set of control signals when a new and persistent pattern emerges. This might be schematized as shown in Figure 2. The equations for the network are then

$$\underline{s}^{(1)} = \underline{M}^{(1)} \underline{x}$$

$$\underline{s}^{(2)} = \underline{M}^{(2)} \underline{x}$$

$$M^{(1)}(0) = p_0 p_0^T + p_1 p_1^T + \dots + p_M p_M^T, \quad M^{(2)}(0) = 0$$

$$\dot{M}^{(1)}(t) = 0, \quad \dot{M}^{(2)}(t) = -\nu M_{ij}(t) + \rho \underline{x}(t) \underline{x}(t)^T$$

The dynamics of STM are as before. Note that in this scheme both LTMs share the same STM. There are many other possibilities, however.

### Stability of the Neural Net

Stability is dealt with in the proposed neural net model in two ways. First, in equation (2) we note that the terms  $(B - x_1(t)) [f_1(x_1(t)) + u_1(t)]$  add to  $x_1(t)$  heavily when  $x_1(t) \approx 0$  but when  $x_1(t) \approx B$  the contribution of this term is zero. Since this is the only term which contributes to  $x_1(t)$ ,  $x_1(t)$  is bounded above by the saturation level  $B$  [Grossberg, 1980].

The other factor which enforces stability is the use of inhibitory signals which prevent STM from saturating at level  $B$ . Each node  $v_i$ , characterized by a state component  $x_i(t)$ , "sees" the outside environment from its own point of view, which prefers some types of signals or some weighted combination of signals. The net as a whole attempts (a) at each node  $v_i$  to excite activity  $x_i(t)$  in proportion to the content  $u_i(t)$  in the input signal vector  $\underline{u}(t)$  to which  $v_i$  is preferentially tuned; (b) at each node  $v_j$ ,  $j \neq i$ , to inhibit activity  $x_j(t)$  in proportion to the content  $u_i(t)$  of the input vector and the recent history of  $x_i(t)$ , i.e.  $f_i(x_i(t))$  shunted by the activity  $x_j(t)$ .

The excitation of a node and inhibition at all other nodes is the "on-center off-surround" action seen from the point of



view of one node. This simultaneous activity going on at each node results in a competitive struggle in which some subset  $(x_k(t))$ , of  $(x_i(t))$  will grow and remain strong and the remaining  $(x_i(t))$  will dwindle and fall below some threshold. This arrangement will prevail for as long as some input pattern persists. This results in a vector  $\underline{x}(t)$  in STM which represents a pattern that is global to the net and which can now be compared with initial patterns  $(p_m)$  in LTM via Equation (3) or any variants thereof [Cohen and Grossberg, 1983].

### Modularity and Neural Nets

Neural nets can generate outputs which behave either as excitatory or inhibitory inputs to other neural nets. It is therefore relatively easy to use a modular approach to organizing a large-scale neural net. In fact the Grossberg outstar configuration [Grossberg, 1978, 1981] allows an entire net to be excited or inhibited by the use of a single control signal. Thus, the net as a whole is capable of implementing switching functions or attention focusing mechanisms which might inhibit some subnetwork activity and excite activity elsewhere e.g., to conserve resources. This is an important consideration for large-scale systems operating in an environment where resources are limited.

### Neural Model Simulator Experimental Implementation

Optical processing/computing is under investigation by the University of Dayton Research Institute (UDRI) and other research organizations in recognition of its perceived natural strengths, which are: (a) speed, (b) parallelism and (c) capability for dense interconnections [IEEE Proceedings, July 1984]. Optical computing approaches may best be used initially as a tool to solve the matrix algebraic model equations which are most computationally intensive. Ultimately, device realizations such as optically interconnected arrays of optical nonlinear devices

(e.g., threshold logic elements) have great potential for implementing large, richly connected networks having neural characteristics.

For the neural simulator suggested here, optical processing techniques and hardware primarily are envisioned in a service role to implement a major portion of a laboratory test bed to be used to investigate adaptive neural networks. Optics will be used to speed up the simulator by performing matrix algebraic operations much faster than could digital equipment of comparable cost and size. The ultimate goal of an all-optical neural processor should not be neglected, although such an implementation is viewed as premature at this time.

The suggested simulator design is a hybrid (optical/digital) system in which the parts are assigned duties most appropriate to their respective natural strengths. The concept of modeling or implementing neural networks and associative memories using optical processing systems recently has been addressed both theoretically and experimentally [Farhat, 1985a and 1985b, Fisher, 1984, Hecht-Nielsen, 1982]. The suggested design is related in subject but comprises an innovative approach based essentially on an extension of the work referenced rather than the Hopfield neural network model [Grossberg, 1976a, 1983; Hopfield, 1982].

The form of the suggested system is shown schematically in Figure 3. Numerous simple optical components such as cylindrical and spherical lenses have been omitted from the figure for the sake of clarity. In all cases the omitted elements are being used in straightforward ways to perform imaging or collimation, with parameters well within their performance capabilities. Operation of the optical part of the system to model Equations (2) and (3) of the proposed neural network will now be described. The system uses a modification of a well known matrix-vector multiplication architecture [Goodman, 1978]. Spatial light modulators (SLMs) are a key real-time device requirement for optical

processing, and may be viewed as electronically programmable masks or transparencies. The updating of the  $M$  matrix stored in SLM2 per Equation (2) is done using the optical train including LD1, SLM2, BS1, D1, LD2, and SLM3. The current  $M$  values in SLM2 are modulated onto a uniform beam and imaged through beamsplitter BS1 to 2-D detector array D1. The update term of Equation (2) is written by the electronic processor into SLM3 and imaged with a bounce off BS1 onto D1 in proper element registration with the image from SLM2. Element-by-element addition occurs by the natural intensity integration of the D1 detector elements which are read out, properly scaled, and formed as the next array  $M$  in SLM2. The other mode of the optical system implements Equation (3) using a well-known matrix-vector multiplication architecture [Goodman, 1978]. The optical train for this mode is: LD1, SLM1, SLM2, BS1, and D2. Vector  $x$  is input to SLM1 (e.g., an acousto-optic cell illuminated by laser diode LD1) and is properly connected to the modulator elements in SLM2 (vertically imaged and horizontally broadcast). After a bounce off BS1, the desired output vector elements are collected on linear detector array D2 by optics which image horizontally and collect vertically. The detailed design of the optical architecture requires further study, and Figure 3 should be considered an example only. It may prove advantageous, for example, to perform the matrix update electronically, since only addition is involved.

Issues such as the representation of bipolar values in the intrinsically unipolar incoherent optical systems, (and, if binary SLMs are used, the processing of analog values) must be considered. Figure 4 is an example of a modification of the matrix-vector multiplier architecture [Goodman, 1978] of Figure 3 which handles bipolar values in both the vectors and matrix and also performs two parallel matrix-vector multiplies using both time sequencing and spatial partitioning. Two parallel multiplies would be used, for example, to model the interaction of two associative memories. Two LTM matrices,  $A(1)$  and  $A(2)$ , are stored in

two-dimensional light modulator SLM2, which is partitioned vertically (parts A and B) to hold the two separate matrices. Horizontal (+ and -) partitioning separates the positive and negative portions of the matrices. The linear SLM, SLM1, is used to input  $\underline{x}$  vectors and is also partitioned vertically (A and B parts) to correspond to the two LTM matrices stored in SLM2. The four-part sequence to perform one complete cycle of operation is as follows: Input modulator SLM1 is first loaded with the positive values of the first input vector  $\underline{x}_1$  in its A section and zeroes in its B section. A matrix-vector multiplication is performed optically with results appearing on linear detector D1. Since the B section of SLM1 contains zeroes, the resulting vector is the product of the A sections of SLM1 and SLM2. Because of the polarity (+/-) partitioning of SLM2, the output vector is presented in polarity-partitioned format on D1. D1 is twice the length of a bipolar vector and contains vector components corresponding to positive and negative contributions. These contributions to the A result are read out of D1 by the electronics and stored in digital memory. Next, the negative elements of input vector  $\underline{x}_1$  are loaded into the A part of SLM1 and the matrix-vector cycle is repeated. A second set of polarity-partitioned result vector components is obtained, this time in reversed polarity order. These are read out and electronically added to the first results to obtain the complete bipolar vector result for Section A. An identical two-step sequence using the B parts of the modulators results in the second vector result.

The electronic system consists of processing, timing, and other circuitry to interface with the electro-optic components. Processing is needed to implement Equation (1) and to perform scaling and other housekeeping operations required by signals transmitted to and received from the SLMs and detector arrays, respectively. The processing load required to implement Equation (1) will vary widely depending on the form selected for  $f(w)$ . Initially, forms which result in a scalar equation (i.e., rela-

tively low processor workload) will be investigated, but it is anticipated that forms requiring vector-matrix operations also will be of interest. Nevertheless, the choice of Equation (1) for digital rather than optical implementation is justified based on the requirement for flexibility to alter the form of Equation (1) (e.g., form of  $f(w)$ ) since flexibility is difficult to implement optically. The optical part of the system is used for matrix-vector operations and benefits from the inherent speed, parallelism, and connectivity of optical implementations.

Some discussion of the hardware requirements implied by the types of systems proposed (Figures 3 and 4) is in order, since they involve real-time devices. By far the most demanding device requirement involved is that of the SLMs, since this device technology is not generally at the off-the-shelf stage. SLM1 can be implemented with an acousto-optic cell with very low risk. All the remaining optical components can be implemented with off-the-shelf commercially available items, many of which are already on hand. In addition to the acousto-optic cell, which can serve as a 1-D SLM, two promising types of 2-D SLM (among others) might be used: the Litton/Semetex magneto-optic LIGHT-MOD™ device [Ross, 1982], and an electronically addressed twisted nematic liquid crystal modulator obtained by modifying a consumer pocket TV unit [Fisher, 1985]. The LIGHT-MOD devices are 48-by-48 element arrays which have the limitation of binary modulation. Also commercially available are 128-by-128 element LIGHT-MODs. The probable maximum size array to be modeled will be about 6-by-6, which allows a 4-by-4-bit array on the LIGHT-MOD to be assigned to each unipolar element of the model, thus enabling techniques such as area weighting to be used to obtain analog modulation. Some level quantization noise is expected using a 4-by-4-element scheme. Due to the anticipated robust nature of neural networks, we expect that the resulting performance degradation will be small. This in itself is an interesting area of research, since many schemes might benefit in terms of practical hardware

requirements if quantization could be tolerated. The LIGHT-MOD devices can also be operated in a duty cycle modulation mode, in which analog signal values are represented by the fraction of time that a modulating element is turned on during a cycle. This mode has been successfully used by LITTON Data Systems, developers of the device, to demonstrate gray scale operation at TV frame rates using a 128-by-128 element device as a display. Both the area weighting and duty cycle modulation methods require a price in drive circuit and/or software complexity. An assessment of this tradeoff should be part of an initial design study as should consideration of the use of other type SLMs such as the liquid crystal (TV) unit. The liquid crystal devices have not been fully characterized; however, they are known to have 120-by-140-element resolution, operate at TV frame rates, and have analog modulation capability. We have measured contrast ratios of 10:1 on these devices. Obviously, 2-D SLMs can be under-utilized as a linear array for SLM1 if expedient. Other state-of-the-art devices, such as the Texas Instruments Deformable Mirror Device, may also be available.

The exact form and detailed design of the electronic portions of the hybrid system requires further definition, but Figure 5 presents a block diagram which indicates the main components. Signal flow is circular through the optical and electronic processors, with each sub-system performing tasks which it does best, as mentioned above. Specialized interface circuits involving functions such as A/D and D/A conversion, amplification, and sample-and-hold are required in the signal flow paths between the optical and electronic domains. In addition, dedicated circuits providing timing, control, and data buffering (as indicated) will probably be required. The electronic processor could be a 16- or 32-bit bus-oriented microprocessor system, using for example, the 68000 chip. This processor will have both real-time functions during simulation runs and other general-purpose duties including software development, data ana-

lysis, and system executive functions. The off-loading of real-time data buffering and timing control functions, using custom circuits (as indicated) is expected to be mandatory to enable the electronic processor to handle the computational load implied by simulation at attractive real-time rates. Mass storage is necessary both for program development and storage and for storage of simulation data and case descriptions.

### References

1. Cohen, M. and S. Grossberg, "Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks", IEEE Trans. Sys., Man. and Cybernetics, SMC 13, 815-82 (1983).
2. Ellias, S. and S. Grossberg, "Pattern Formation, Contrast Control, and Oscillations in the Short Term Memory of Shunting On-Center Off-Surround Networks", Biol. Cybernetics, 20, 69-98 (1975).
3. Farhat, N. and D. Psaltis, "Optical Information Processing Based on an Associative-Memory Model of Neural Nets with Thresholding and Feedback," Optics Letters, 10 (2), 98-100 (1985a).
4. Farhat, N., D. Psaltis, A. Prata, and E. Paek, "Optical Implementation of the Hopfield Model," Applied Optics, 24 (10), 1469-1475 (1985b).
5. Fisher, A., Ad Hoc remark delivered at the "Optical Society of America Topical Meeting on Optical Computing," Incline Village, Nevada, 18-20 March 1985.
6. Goodman, J. W., A. Dias, and L. Woody, "Fully Parallel, High-speed Incoherent Optical Method for Performing Discrete Fourier Transforms," Optics Letters, 2 (1), 1-3 (1978).
7. Grossberg, S., "Adaptive Resonance in Development, Perception and Cognition," SIAM-AMS Proc., 13, 107-156 (1981).
8. Grossberg, S., "Contour Enhancement, Short Term Memory, and Constancies in Reverberating Neural Networks," Appl. Math. 52 (3), 213-257 (1977).
9. Grossberg, S., "Adaptive Pattern Classification and Universal Recoding: I. Parallel Development and Coding of Neural Feature Detectors," Biol. Cybernetics, 23, 121-134 (1976a).
10. Grossberg, S., "Adaptive Pattern Classification and Universal Recoding: II. Feedback, Expectation, Olfaction, Illusions," Biol. Cybernetics, 23, 187-202, (1976b).
11. Grossberg, S., "How Does A Brain Build a Cognitive Code?", Psychol. Review, 87 (1), 1-51 (1980).
12. Grossberg, S., "A Theory of Human Memory: Self Organization and Performance of Sensory Motor Codes, Maps and Plans". In R. Rosen and F. Snell (Eds.), Progress in Theoretical Biology, Vol. 5, New York: Academic Press, 1978.



13. Hecht-Nielsen, R., Neural Analog Processing, SPIE 360, 180-189 (1982).
14. Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. Natl. Acade. Sci., U.S.A. 79, 2554-2558 (1982).
15. IEEE Proceedings, "Special Issue on Optical Computing," 72 (7) 753-992 (1984).
16. Kohonen, T., Self Organization and Associative Memory, (Springer-Verlag, New York, 1983).
17. Kohonen, T., E. Oja, and P. Lehtio, "Storage and Processing of Information in Distributed Associative Memory Systems," Parallel Models of Associative Memory, ed. Hinton, G. and Anderson, J., (Lawrence Erlbaum Associates, Hillsdale, NJ, 1981).
18. Ljung, L. and Soderstrom, T., Theory and Practice of Recursive Identification, (Cambridge, MA: The MIT Press, 1983).
19. Monzingo, R. A. and Miller, T. W., Introduction to Adaptive Arrays, (New York: John Wiley and Sons, 1980).
20. Papoulis, A., Probability, Random Variables, and Stochastic Processes, (New York: McGraw-Hill, 1984).
21. Ross, W. E., D. Psaltis, and R. Anderson, "Two-Dimensional Magneto-Optic Spatial Light Modulator for Signal Processing," SPIE 341, "Real-time Signal Processing V," 191-198 (1982).
22. Willshaw, D., "Holography, Associative Memory, and Inductive Generalization," Parallel Models of Associative Memory, ed. Hinton, G. and Anderson, J., (Lawrence Erlbaum Associates, Hillsdale, NJ, 1981).

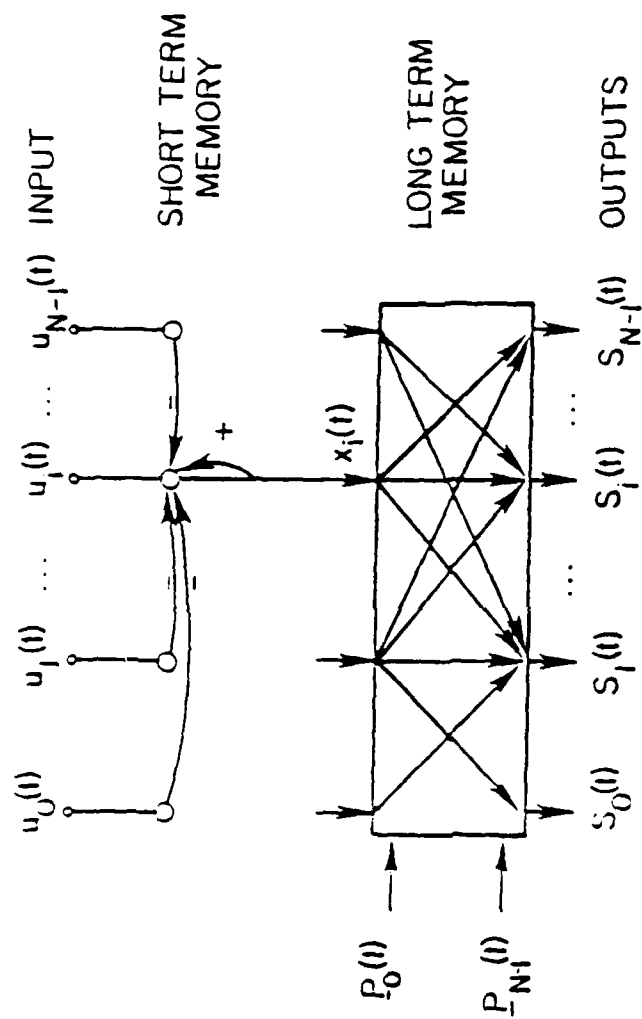


Figure 1. Network Model to Perform Adaptive Pattern Recognition with Monitoring.

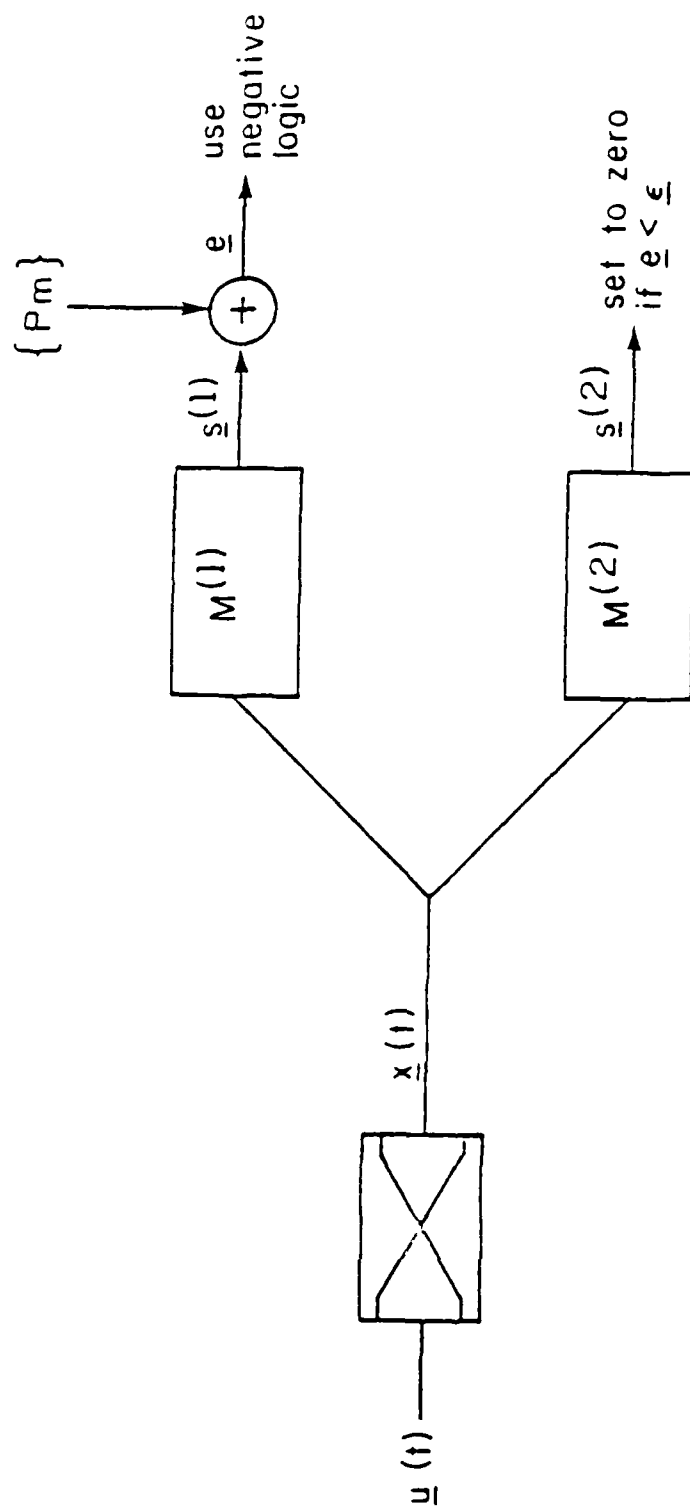


Figure 2. Example Network with 2 LTMs.

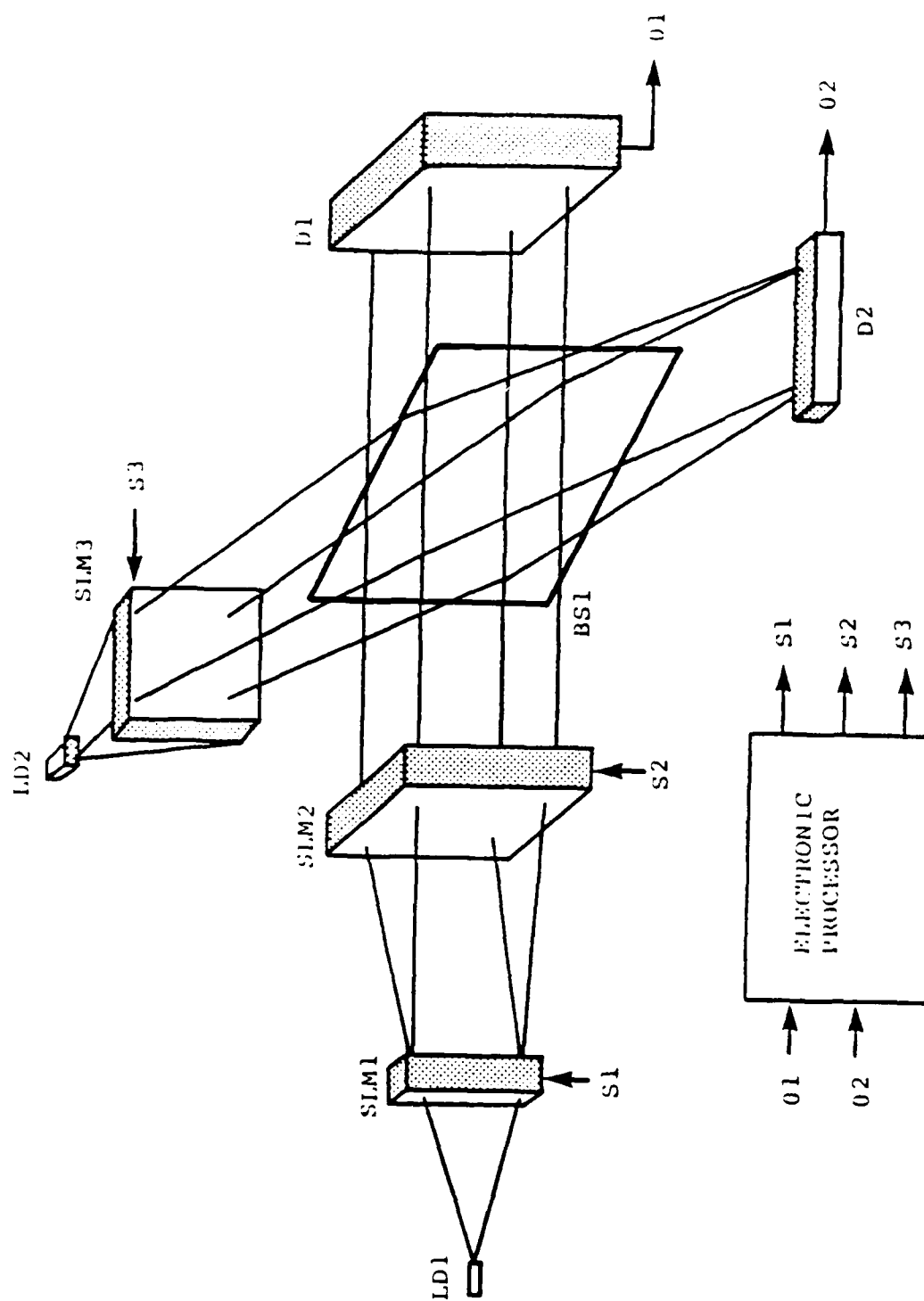


Figure 3. Schematic of Experimental System.

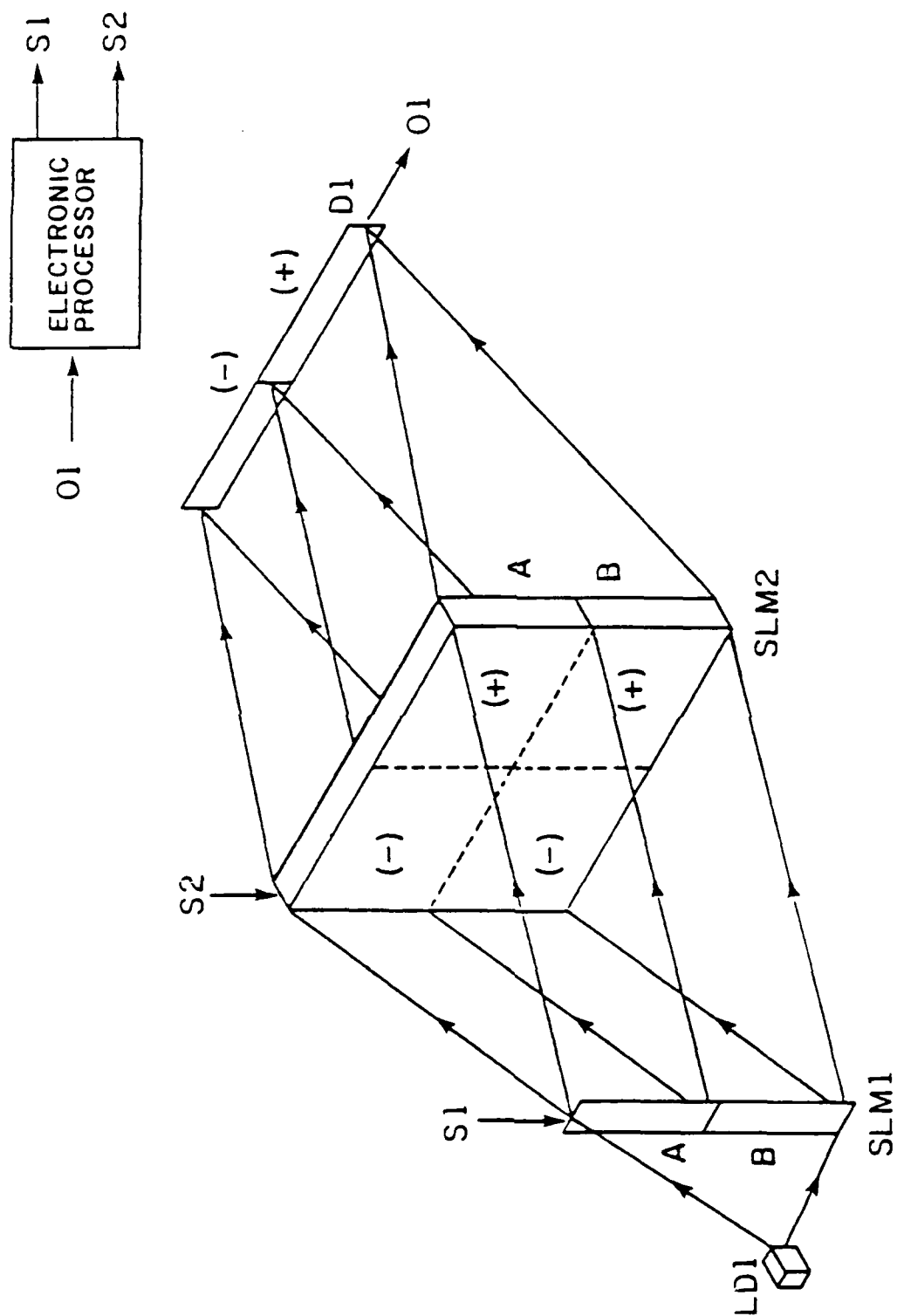


Figure 4. Architecture for Performing Two Matrix-Vector Multiplications with Bipolar Values.

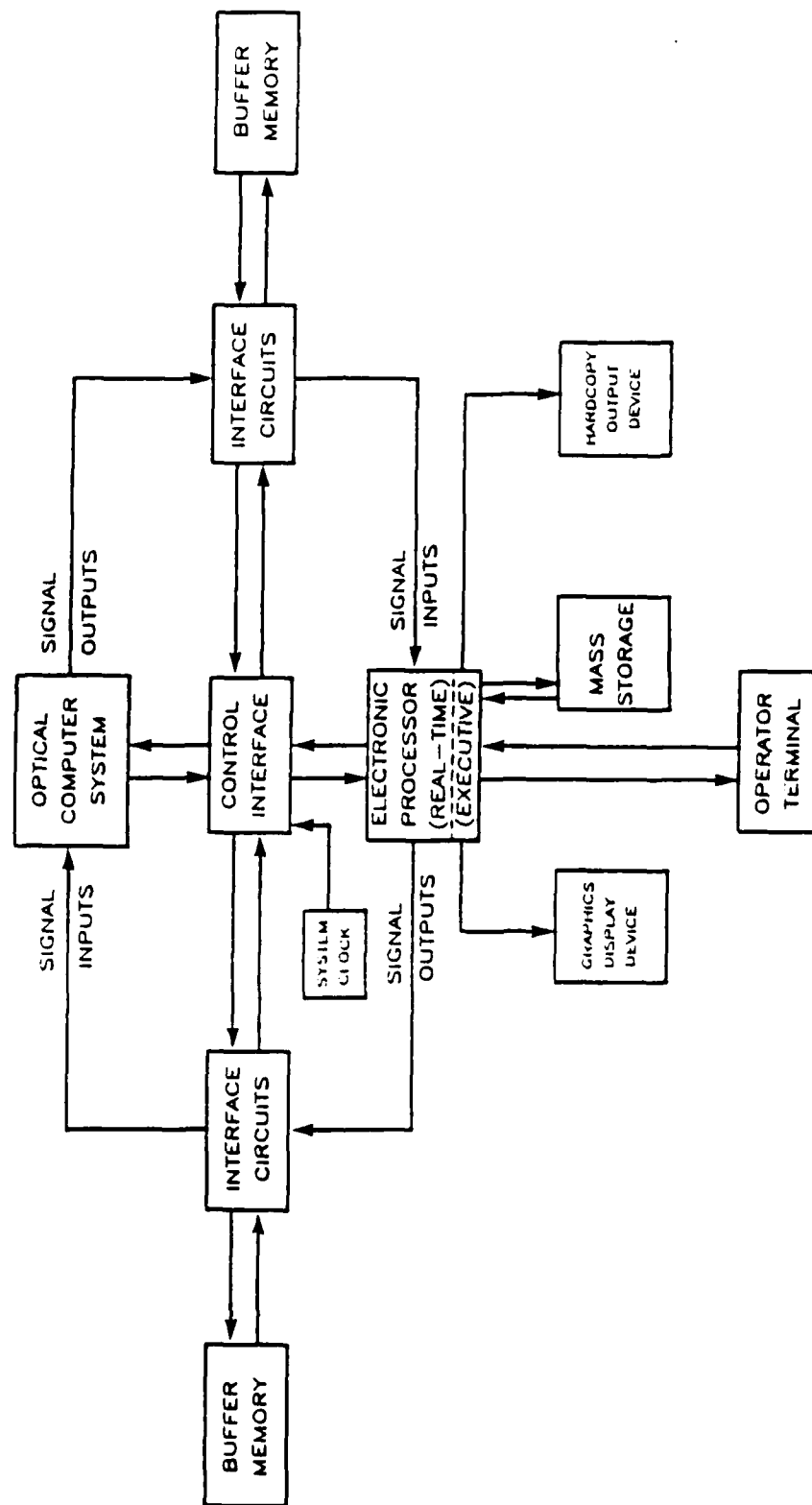


Figure 5. Block Diagram of Experimental Neural Network Simulator System.

NOTE ON COHERENCE THEORY  
AND OPTICAL PROCESSING SYSTEMS

Steven L. Cartwright  
University of Dayton Research Institute  
Dayton, Ohio 45469

Optical processing systems--particularly analog processors--often depend on coherent light to perform mathematical operations. The assumption of full spatial coherence leads to a description of optical propagation which is linear in the field amplitude and which relates conjugate planes through Fourier transforms. The assumption of temporal coherence (monochromaticity) eliminates wavelength scaling effects and greatly simplifies the system description. The "classical" coherent optical processor then uses laser illumination, a two-dimensional input in the form of a transparency, various lenses to create the Fourier transform of the input, means of manipulating the Fourier transform (usually a hologram), additional lenses to reform the filtered image, and a two-dimensional output sensor (usually a television camera). The criticisms of such a system are by now well known, including the system's inflexibility and limited accuracy. The standard solutions to these problems are equally well known, including the use of two-dimensional spatial light modulators for the input and Fourier plane filters. These solutions compound the accuracy problem by limiting the resolution of the system to several dozen wavelengths. The price paid in component cost and power consumption is also substantial.

In an attempt to circumvent the accuracy and flexibility problems of analog processors, the optical processing community has turned to digital processors, as did the electronic processing community years ago. In a digital optical processor the input and output are abstract numbers, the magnitudes of which are represented by light intensity levels. Even if the input representation is binary, all optical computers proposed so far

have a multilevel output, referred to as mixed radix representation. Practical implementations of optical computers require modulating devices which are inefficient. To overcome this problem recent attention has centered on bistable optical devices which exhibit thresholding and gain. A more subtle drawback to optical computing is that all systems proposed so far are based on a geometrical optics design approach. Since geometrical optics is only an approximation of how light propagates, these optical computers encounter unanticipated difficulties when put into practice, particularly from diffraction effects.

A fruitful approach to using optics for computing and signal processing is to gain a deeper understanding of how an optical system works and to use this understanding to integrate optics into electronic processors. Optimum optical processing system designs or design tradeoffs might be identified through a thorough study of optical propagation from a coherence theory viewpoint with an emphasis on optical processing applications. The cross-spectral density function<sup>1</sup> has recently been used as a tool to analyze imaging and processing systems.<sup>2</sup> The cross-spectral density function has the advantage of being perfectly general in both spatial and temporal coherence. It is also equivalent to other recent analyses of generalized imaging systems.<sup>3,4</sup> The philosophy behind designing optical processors from a coherence viewpoint assumes that coherence is introduced into an optical system through a generalized grating, the standard Ronchi ruling being the simplest type. With proper shaping, the coherence function may be made to sample an input or carry information. A second grating combines mutually coherent points to establish the processing operation. We present here two possibilities of how a thorough knowledge of coherence theory would enhance the design of optical processors and computers. The first example is an analog processor, and the second example is a digital processor.



The most common input for an analog optical processor is a two-dimensional image. In a coherent processor the image must be converted to a transparency and illuminated by a laser. In an incoherent system the image is manipulated directly (and passively) by appropriately placed gratings and apertures. No spatial light modulators or extra illumination sources are needed, resulting in a great savings in cost, complexity, and power consumption. Systems that perform edge detection have already been described in the literature.<sup>4,5</sup> These systems use simple Ronchi rulings to introduce coherence in one dimension. By using generalized gratings--essentially arrays of pupils--more complicated operations are possible. Because of the broad-band nature of the input the output can be separated by color filters to give differently scaled results in different wavelengths. The filtering operations can be performed at any wavelength for which gratings and detectors exist. In an end application we might have an infrared imaging system with a sophisticated lens on the front end. The lens would perform a specific operation which would pre-filter the image before it reached the detector, thus simplifying the electronic processor. Because of the broad-band illumination in such a system we need the cross-spectral density function analysis to be able to predict the system performance. In addition, the cross-spectral density function leads to an analysis which, though complicated, can lend itself to simple interpretation.

In current thinking, a digital processor or optical computer would contain several bistable devices to perform thresholding. To take advantage of the two-dimensional nature of optical propagation there would likely be two-dimensional arrays of these devices. There might also be two-dimensional arrays of laser diodes used as light sources. Although each diode would in itself be spatially coherent, each diode would be incoherent with respect to its neighbors. Because they are intensity dependent devices, each bistable device in an array would be incoherent

with respect to the other devices. Thus, a digital optical processor would essentially be a spatially (but not temporally) incoherent system. As light propagates through the system, information from each cell of the bistable array is mixed with the other cells. This mixing, if it can be understood and controlled, can be used to form interconnections between cells. A coherence analysis would show how best to take advantage of the effects of non-coherent propagation. By introducing generalized gratings and pupils (possibly programmable) the operation of the optical computer could be enhanced.

### References

1. L. Mandel and E. Wolf, "Spectral Coherence and the Concept of Cross-Spectral Purity," J. Opt. Soc. Am., 66, pp. 529-535, June 1976.
2. S. Cartwright, "Incoherent Optical Processing: A Coherence Theory Approach," Appl. Opt., 23, pp. 318-323, January 1984.
3. G. J. Swanson and E. N. Leith, "Analysis of the Lau Effect and Generalized Grating Imaging," J. Opt. Soc. Am. A., 2, pp. 789-793, June 1985.
4. G. Indebetouw and C. Varamit, "Spatial Filtering with Complementary Source-Pupil Masks," J. Opt. Soc. Am. A., 2, pp. 794-798, June 1985.
5. W. Stoner, "Incoherent Optical Processing via Spatially Offset Pupil Masks," Appl. Opt., 17, pp. 2454-2467, August 1978.

STABILIZATION AND CONTROL TECHNIQUE  
WITH IMPROVED PRECISION

Gordon R. Little  
University of Dayton Research Institute  
Dayton, Ohio 45469

Stabilization and control of the phases of mutually coherent optical sources are important in many areas of modern optics, including holography,<sup>1-4</sup> and coherent optical computing.<sup>5</sup> Over the years several active schemes for achieving phase stabilization and control have been reported.<sup>1-5</sup> The method of MacQuigg<sup>3</sup> is relatively easy to implement but yields limited phase control while that of Rakhimov and Tronko<sup>4</sup> requires the use of magneto-optic gratings and imposes overly strong restrictions on the source geometry. We report on a simple modification of MacQuigg's method which yields good phase control in an easily implemented system.

The operating principles of MacQuigg's method are illustrated in Figure 1. Here, it is desired to stabilize and control source  $S_2$  relative to source  $S_1$ . A simple phase control grating is fabricated by overlapping the two beams and making a hologram. When the hologram is placed in its original position and rotated through a small angle about an axis parallel to the grating lines, straight equally spaced fringes will appear in the two beams leaving the grating. These fringes are due to the interference of the  $S_1$  and  $S_2$  beams with the tilted reconstructed waves. A detector is placed in one of the beams and its output is measured with a lock-in amplifier. The oscillator output from the lock-in amplifier is used to drive a phase shifter on  $S_2$  providing a small dither in the fringe positions. Since the lock-in amplifier output is zero when the fringe pattern is at either a maximum or a minimum on the detector, this output can be used as an error signal in a closed loop feedback system, thereby stabilizing the phase of  $S_2$  relative to  $S_1$ .

Phase control in this scheme was achieved by translating the grating in the direction perpendicular to the fringe orientation. However, since the grating spacing is of the order of the source wavelength, accurate phase control requires extremely high resolution in the translation device. For example, with the 0.488  $\mu\text{m}$  Argon ion laser line and a source angular separation of  $10^\circ$ , a grating translation of 0.08  $\mu\text{m}$  corresponds to a  $10^\circ$  phase shift. Clearly, phase control of  $1^\circ$  would be difficult using this approach.

To achieve greater accuracy in phase control we suggest translating the feedback loop detector rather than the grating. Since the fringe spacing can easily be adjusted to be relatively large by controlling the grating tilt, and since translation of the detector through one fringe corresponds to  $360^\circ$  phase change, the resolution in phase control can be much greater, independent of the actual grating spacing. Calibration of the phase control is achieved by monitoring the fringe in the second fringe pattern using an auxiliary detector. A typical calibration is shown in Figure 2 where the output I of the monitoring detector ( $\Delta$ ) is plotted as a function of the position of the feedback loop detector. Also shown (solid curve) in the figure is the best fit curve of the form

$$I = a_4 + a_3 \sin 2\pi a_2(x - a_1) \quad .$$

In this case,  $a_2$  was found to be  $.2718 \pm 0.0001$  cycle/mm yielding a phase control calibration factor of  $97.85 \pm 0.04^\circ$  mm. A simple micrometer-driven translation stage with 0.01 mm resolution could easily yield  $1^\circ$  phase setability. It should be noted however, that it is the mean phase which can be controlled to this level. Short term phase fluctuations, due to the dither for example, will be considerably larger. In many applications, such as in fabricating holograms for use in coherent optical processors, the short-term fluctuations will act only to slightly reduce the hologram contrast.

A simple variant of the detection scheme yields a more efficient use of the available optical power. Here a mask featuring slit apertures that match the fringes is placed in the pattern and a lens is used to focus all of the passed radiation onto the detector. The mask, rather than the detector, is translated to achieve phase control.

To stabilize and control multiple sources, the phase control hologram can be fabricated using multiple, non-overlapping exposure of pairs of the sources. For  $N$  sources ( $N - 1$  to be controlled relative to the first), there will be  $N$  fringe patterns. The one associated with the reference source will contain  $N - 1$  fringe pattern segments while the others will each have one segment. The  $N - 1$  single segment patterns can be used in feedback loops for stabilization and control while the composite can be used for calibration. Photographs of the fringe patterns for a four-source system are shown in Figure 3. The different spacings of the fringes are due to the differing angular separations of the sources.

## REFERENCES

1. Don B. Neumann and Harold W. Rose, "Improvement of Recorded Holographic Fringes by Feedback Control," Appl. Opt. 6, 1097 (1967).
2. H. W. Rose and H. D. Pruett, "Stabilization of Holographic Fringes by FM Feedback," Appl. Opt. 7, 87 (1968).
3. David R. MacQuigg, "Hologram Fringe Stabilization Method," Appl. Opt. 16, 291 (1977).
4. D. A. Rakhimov and V. D. Tronko, "Stabilization of the Optical Path of a Holographic Platform," Opt. Spectrosc. (USSR) 48 (2), 227 (1980).
5. Clark C. Guest and Thomas K. Gaylord, "Phase Stabilization System for Holographic Optical Data Processing," Appl. Opt. 24, 2140 (1985).

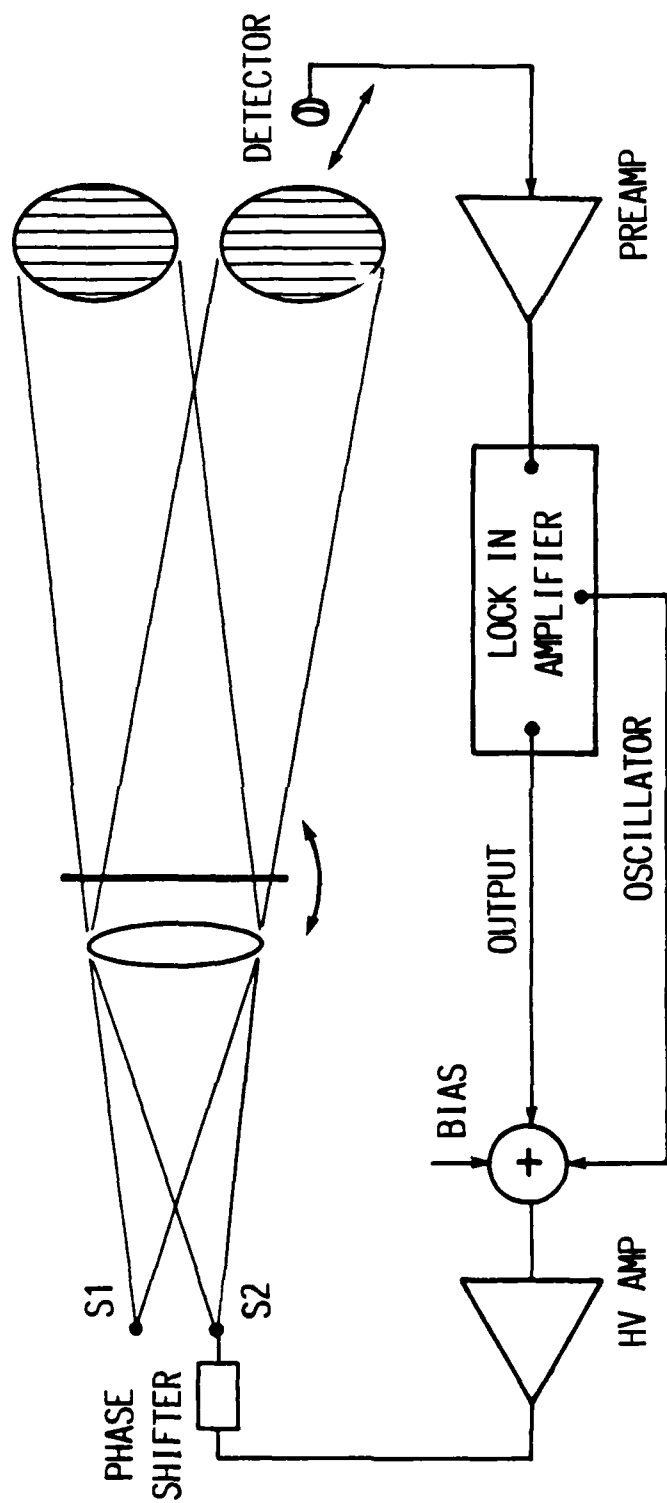


Figure 1. Phase Stabilization and Control Scheme of MacQuigg.3



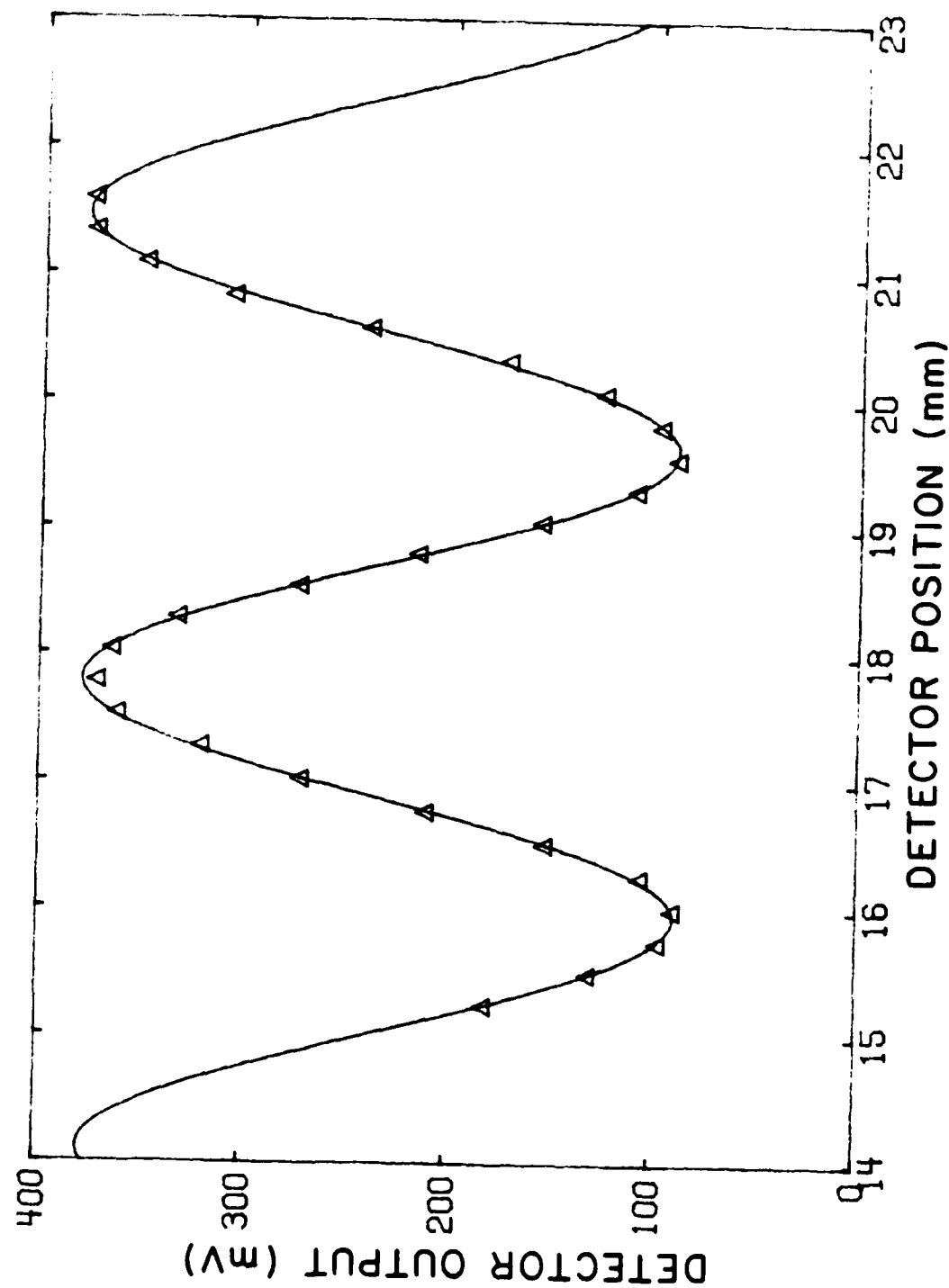


Figure 2. Calibration of fringe control system. The points marked by triangles designate measured data while the solid curve is the best-fit curve.



Figure 3. Photograph of the fringe patterns in a system to stabilize and control three sources relative to a fourth source.

INTEGRATED-OPTICAL DEVICES FOR NONLINEAR COMPUTATION

Richard P. Kenan and Carl Verber, Principal Investigators

BATTELLE  
COLUMBUS LABORATORIES

FINAL REPORT

on

INTEGRATED-OPTICAL DEVICES FOR  
NONLINEAR COMPUTATION

to

UNIVERSITY OF DAYTON RESEARCH INSTITUTE  
300 College Park Avenue  
Dayton, Ohio 45469

from  
BATTELLE  
COLUMBUS LABORATORIES  
505 King Avenue  
Columbus, Ohio 43201

## TABLE OF CONTENTS

INTRODUCTION . . . . .	3
DISTRIBUTED THRESHOLD COMPUTING . . . . .	3
PIPELINED POLYNOMIAL PROCESSOR . . . . .	3
ALL-OPTICAL A/D CONVERTER . . . . .	4
INTRODUCTION . . . . .	4
BASIC DESIGN . . . . .	4
THE COMPONENT REQUIREMENTS . . . . .	5
ANALYSIS OF NONLINEAR COMPONENTS . . . . .	6
Wave Propagation in Nonlinear Media . . . . .	6
Propagation in Nonlinear Waveguides . . . . .	7
REFERENCES . . . . .	9

## INTRODUCTION

The research efforts reported here cover work done in a six month initial-phase project consisting of design and consulting tasks. Limited work of a theoretical nature was carried out, and the two presentations were given (one presentation at the SPIE Annual Meeting in San Diego; one at the OSA meeting in Washington, DC). No experimental or fabrication effort was planned or carried out.

In the two short segments following, work in Distributed Threshold Computing Implementations and in preliminary design of a Pipelined Polynomial Processor are described. Then, the principal thrust of our work, the All-Optic Analog/Digital Converter, is described in some detail. An integrated-optic layout for this device is presented, and the needed analysis is discussed. A copy of a joint publication (with UDRI staff) is attached as an appendix.

### DISTRIBUTED THRESHOLD COMPUTING

This activity was performed in cooperation with researchers from UDRI; it consisted primarily in looking for ways to implement the architectures of interest to UDRI in integrated-optic format. As this was a support activity, and no device parameters were identified, little detailed design was carried out. There was some design work done to try to devise a way to implement a 2x2 multiplier by direct implementation of the multiplication table in threshold logic. We were successful in finding two possible implementations, both using waveguide horn structures. In the first, the entire logic takes place in the horns; in the second, the horns are meant only to convey the light to a multimode region where collection gratings assemble the output light into the proper directions. This work was dropped because insufficient time for the UDRI researchers to fully develop their concepts and because of attention to the A/D converter described below.

### PIPELINED POLYNOMIAL PROCESSOR

A modest effort in preliminary grating design was made in support of this processor concept. The device is based upon the paper by Verber et al<sup>1</sup>. The device, as envisioned for integrated-optical implementation, utilizes two kinds of optical gratings: (1) an "adder" grating, used as a beam combiner to combine the pipeline mainstream data with new coefficient data; and (2) a multiplier, to be constructed using an electrooptic grating. The relevant features of these gratings are that type (1) is a holographic surface grating with a large deflection and a very narrow angular range of operation, while type (2) is formed photolithographically and therefore has a large period and a wide angular range of acceptance, but a small deflection. For this

report, further details on the device layout and operation are not relevant.

The work done on this project consisted of looking at possible geometries for adder gratings (type 1) and the impact on the area of crystal required for the various geometries. Two geometries were considered: (1) a "crossed-beam" geometry, where the grating occupies only the region of intersection of the beams being combined; and (2) a Kogelnik<sup>2</sup> type of grating, where the grating extends outside of the region of intersection. It was concluded that no strong "real estate" advantages would accrue to either design, but that the Kogelnik type of grating might be easier to fabricate because of looser alignment tolerances as far as the beam paths were concerned.

## ALL-OPTICAL A/D CONVERTER

### INTRODUCTION

By an all-optical analog-to-digital converter, is meant an A/D which accepts an optical input whose intensity is an analog representation of a numerical value, and whose output is an optical binary representation of that analog value. We have chosen, for reasons of compactness, stability, and power consumption, to consider an integrated optical design. It is intended that the integrated optical A/D be a general purpose device which could be incorporated into a variety of systems, although the primary motivation for this work was the need for A/D conversion at the output of an integrated optical DMAC (optical digital multiplication by analog convolution) device. This work led not only to a novel architecture of the optical A/D but also to concepts for waveguide devices which, through the proper use of nonlinear optical materials, could perform the required nonlinear operations.

### BASIC DESIGN

The design of the A/D, as shown in Figure 1, is an adaptation of the electronic flash converter. It consists of:

- An analog input channel,
- Taps for removing equal amounts of energy from the input channel,
- Optical thresholding devices or comparitors which pass a fixed signal when the input exceeds the threshold and no energy when the signal is below the threshold,
- A set of XOR gates,
- A distribution network, and
- A set of binary output channels.

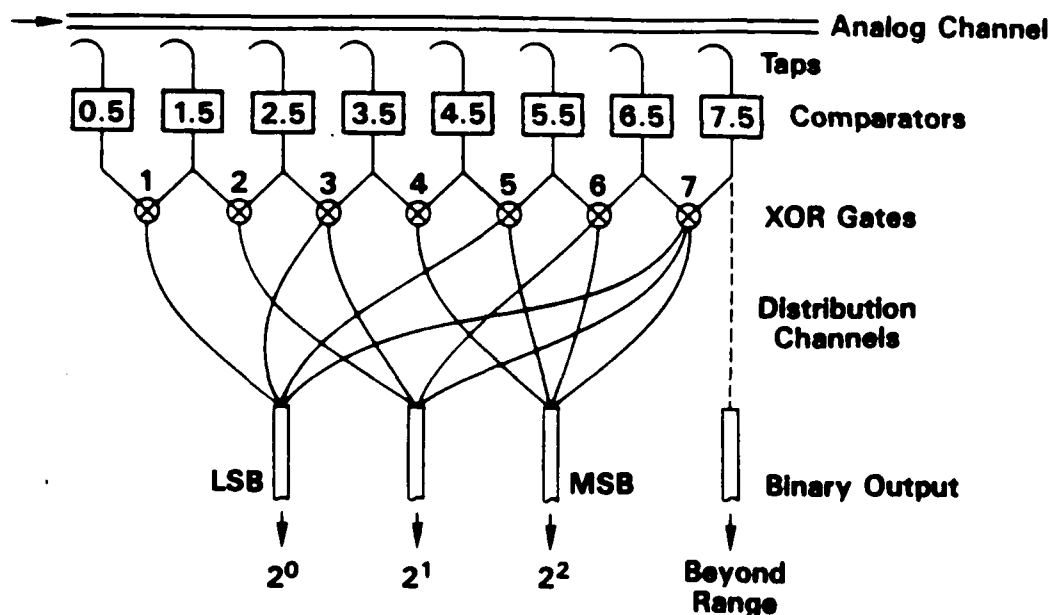


Figure 1. Optical version of the flash A/D converter.

There are a number of factors which must be considered more-or-less simultaneously when considering design alternatives for an I.O. implementation of the A/D:

- Maintain reasonable overall size.
- Provide suitably sharp thresholding action.
- Maintain outputs from the threshold elements consistent with XOR requirements.
- Aim for uniform loss in distribution network.

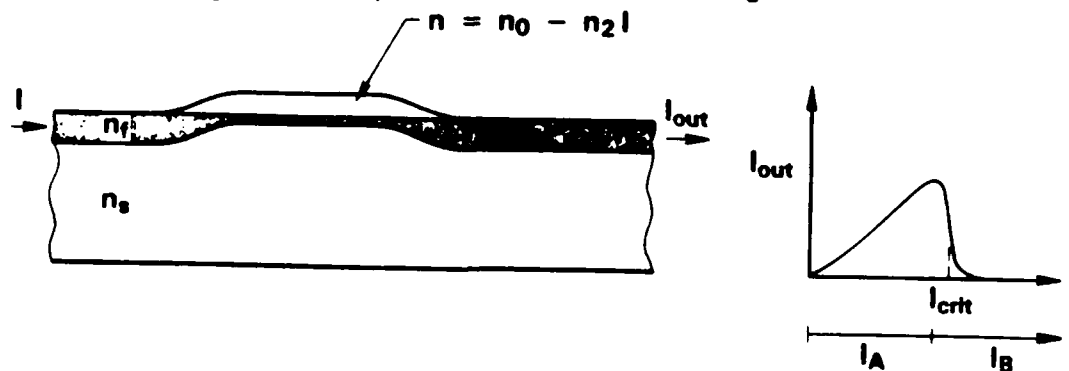
#### THE COMPONENT REQUIREMENTS

The threshold device and the XOR are two "active" devices which must be designed so that they interact properly and so they can be compatibly fabricated on the same substrate. The principal design decision initially was thought to be the form of the threshold device, since this will impact the design of the taps and even the analog input channel. However, it was seen that the design of the XOR has an equal if not greater impact on the overall design.

The best known integrated optical method for implementing the XOR is based upon the operation of the single-mode Y junction<sup>3</sup>, although an equivalent operation can be performed with collimated beams in a planar waveguide using a surface-grating beam splitter. In both cases the XOR operation depends upon the phase difference between the two incident beams. This is a very inhibiting requirement since, in the case of the A/D, it means that there has to be a knowledge of the phase of the light leaving each of the threshold elements.



We arrived at an entirely different design for an XOR gate which was suggested by the work of Seaton, Stegeman and Winful<sup>4</sup> on intensity-dependent guided wave phenomena. They discuss an approach to nonlinear devices in which the nonlinear properties of the waveguide make the modal characteristics intensity dependent. Furthermore, they present data on liquid-crystal MBBA clad glass, III-V materials and ZnS, suggesting that there are a number of materials of which experimental devices might be fabricated. Ignoring, for the present, the details of the materials questions, let us consider Figure 2 which



$$n_f > n_s$$

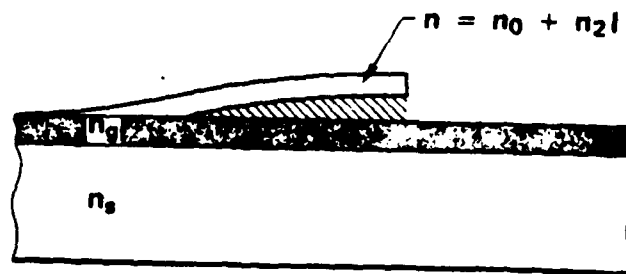
$$n \approx n_f \text{ for } I < I_{crit}$$

$$n < n_f \text{ for } I > I_{crit}$$

Figure 2. Suggested design of an XOR gate using a waveguide overlay with a negative nonlinearity. As shown in the insert, a single input of intensity  $I_A$  passes through the gate but two simultaneous inputs with total intensity  $I_A + I_B$  will be ejected into the substrate.

suggests one way of using this nonlinear waveguide approach to design an XOR. The input waveguide is designed to be a single-mode guided which operates just above cutoff for low light intensities. The overlay material has a negative nonlinearity, that is the index decreases with increasing intensity. As the intensity increases the guide index drops and it cuts off the mode and ceases to transmit. As shown in Figure 3, it is also possible to accomplish the same function using a material with a positive nonlinearity.

The first approaches considered for the threshold device were variations of the nonlinear Fabry-Perot interferometer<sup>5</sup>. However, we found that it is, in principle, possible to perform this function using a variant of the nonlinear waveguide. The basic threshold device is shown in Figure 4. It is designed so that the channel is cut off at low light intensities. Since the nonlinearity is positive, at high intensities the channel can be driven above cutoff. As can be seen, the output of such a

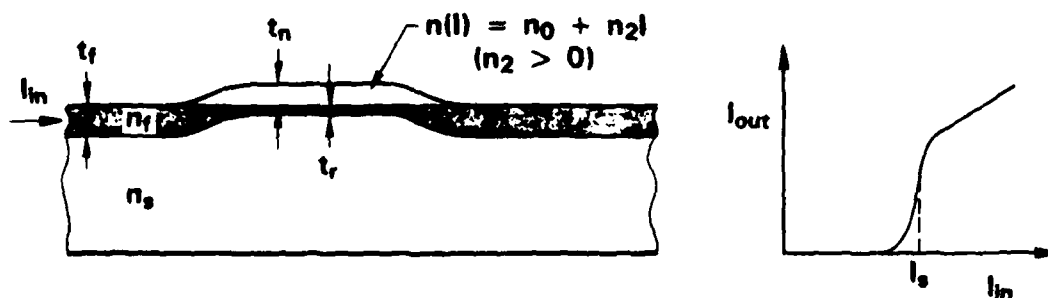


$$n_g > n_s$$

$$n < n_g \text{ for } I < I_{\text{crit}}$$

$$n > n_g \text{ for } I > I_{\text{crit}}$$

Figure 3. Suggested design of an XOR gate using a waveguide overlay with a positive nonlinearity. At a sufficiently high input intensity the guided wave will be lifted out of the waveguide resulting in a characteristic similar to that shown in Figure 2.



Single-mode guide in  $t_r$  region  
 Cut off in  $t_r$  region for low intensities  
 Guides in  $t_r$  region for high intensities

Figure 4. Suggested design of a waveguide threshold device using a waveguide overlay with a positive nonlinearity. As shown in the insert, the output is a linear function of the output when the output is above threshold.

device should go from zero, through a transition region and then become a linear function of the input. It is, of course, desirable that the output of the threshold device be independent

of the input signal strength once the input has exceeded the threshold value.

It will be shown that the proper threshold characteristic as well as other advantages result from the device geometry shown in Figure 5. By separating the output of the device from the

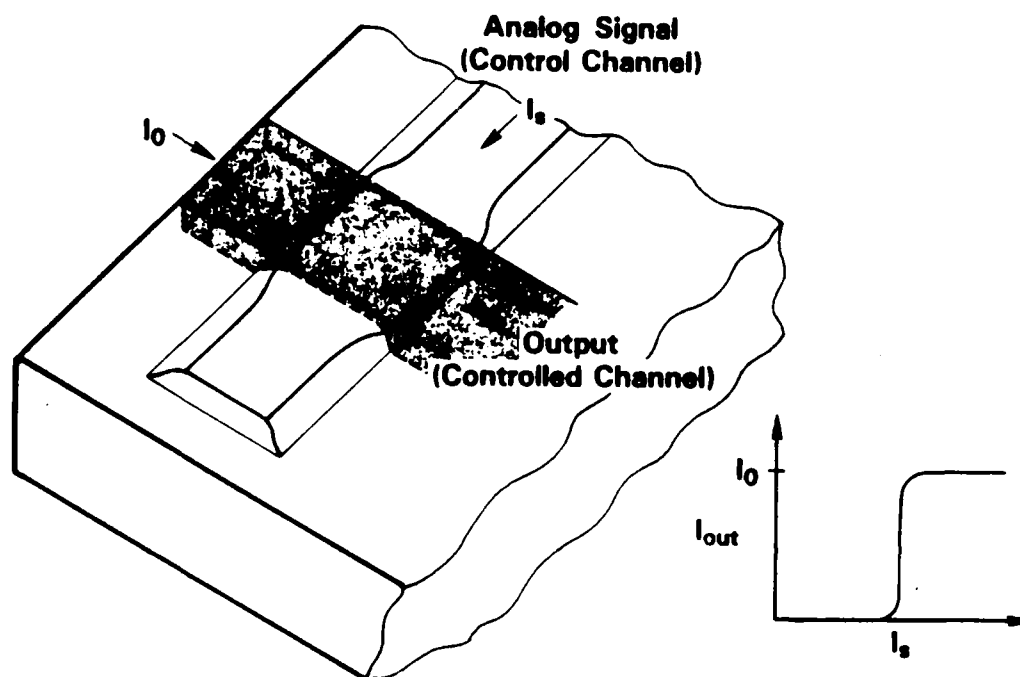


Figure 5. Crossed-channel threshold device geometry. The passive output channel is below cutoff until the signal in the control channel is high enough to "burrow through" the thinned crossing region which is made of a material with a positive nonlinearity. Increasing the index in this region also raises the controlled channel above cutoff and results in the characteristic shown in the insert.

control signal (which in our case is the analog input), we not only achieve the desired gate characteristic, but have allowed a simple solution to the problem of energy distribution at the front end of the device. As shown in Figure 6, the input distribution is accomplished by using an input channel with a finite attenuation which crosses a set of identical threshold units. By the proper logarithmic spacing of these units, we can design them so that they will each turn on at the appropriate value of the input intensity.

$$I_s(y) = I_s(0) e^{-\alpha y}$$

Where  $\alpha$  is the attenuation of the signal channel

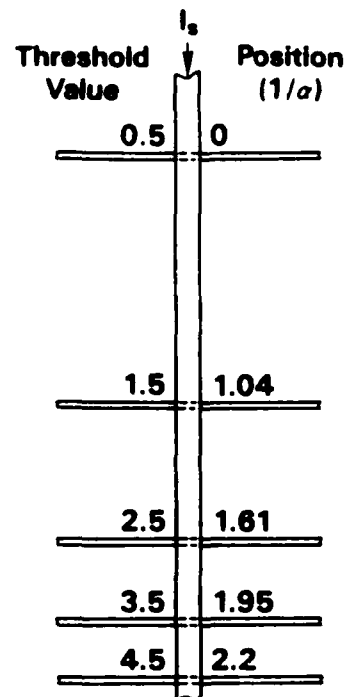


Figure 6. The input signal  $I_s$ , enters in a channel with a fixed attenuation  $\alpha$ . This channel intersects a set of identical threshold devices of the type shown in Figure 5. Since the spacing is logarithmic in the attenuation of the input channel, each threshold unit will turn on at the indicated value even though the units are identical.

A suggested layout for the entire A/D is shown in Figure 7. The outputs of the XOR gates enter a planar waveguide region, and the distribution of the light into the binary output channels is accomplished by the use of holographic optical elements whose design will not be discussed here.

The major development required for the successful implementation of the all-optical A/D is that of the nonlinear elements. This must proceed by a careful theoretical description of the elements which will lead to an ability to predict device performance as a function of the optical properties of the materials considered for the device fabrication. This question is dealt with in more detail in the following section.

#### ANALYSIS OF NONLINEAR COMPONENTS

The nonlinear waveguide components envisioned for use in the Integrated Optic A/D Converter (IOAD) device are a totally new concept, neither analyzed nor constructed in previous work. There are fundamental questions regarding the propagation of light in

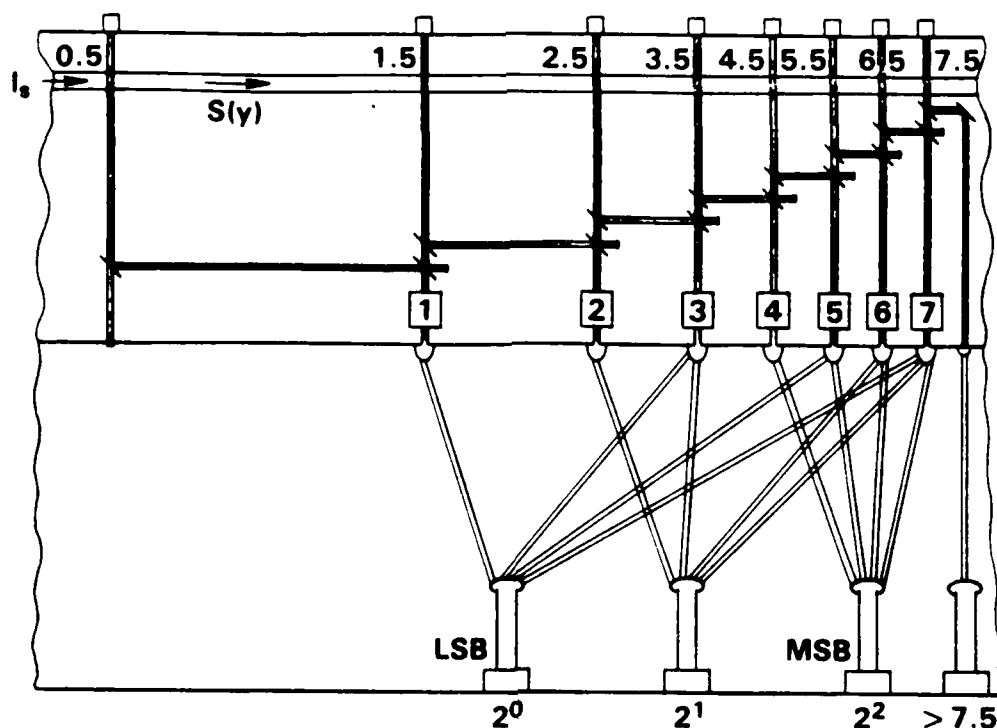


Figure 7. Schematic of the entire optical A/D. Grating beam splitters are suggested to divide the outputs of the threshold elements and to combine the inputs to the XOR gates (which are shown as  $N$ ). This will work properly if adjacent channels are mutually incoherent. As shown in the figure, each channel is fed by its own laser diode.

nonlinear structures, questions that will require extensive analysis to resolve. In this section, we discuss some of the theoretical questions to be answered.

#### Wave Propagation in Nonlinear Media

A number of authors have analyzed wave propagation in nonlinear media; much of this work is reviewed by Akhmanov et al<sup>6</sup>. Among the important results is the instability of homogeneous plane waves in nonlinear media. When the intensity of the waves becomes large enough that nonlinear effects are important, a plane wave may become unstable and break up into an inhomogeneous, filamentary structure.

In linear media, it is often permissible to treat a finite optical beam having a width of many wavelengths as if it were, effectively, a plane wave. In Kerr media, i.e., media having a dielectric constant depending on the intensity of light as

$$c = c_0 + c_2 |A|^2,$$

(1)

this assumption is no longer correct. The spatial transients occurring at the edge of a finite beam are of supreme importance in determining how the beam will propagate. In ref.6, it is shown that such media may develop waveguides by "self action", that is, the finite light beam interacts with the medium to produce a stable index variation that traps the beam.

When an interface with another medium, even a linear one, is introduced, the dynamics of beam propagation become more complex. A number of Soviet authors have published on this subject, one of the earlier ones being Bioko et al<sup>7</sup> who derive expressions for plane waves incident upon a Kerr medium at an interface with a linear medium. For partial reflection, they allow a plane wave to be transmitted into the nonlinear medium [but, they remark that such a wave is unstable]. At total reflection, they use an inhomogeneous plane wave and derive the form of this wave for a semi-infinite nonlinear medium. A series of papers by Rozanov<sup>8,9,10,11</sup> discusses various aspects of the nature of the interactions with nonlinear interfaces. Kaplan<sup>12,13</sup> analyzed hysteresis effects in the reflectivity, using a plane-wave analysis (criticized by Rozanov<sup>10</sup>). These works generated a controversy concerning the behavior of light in a nonlinear medium; the effects of finite beams in contrast to infinite plane waves; and the effect of a single interface on the interactions.

Part of the controversy was resolved in a series of works by Smith et al<sup>14,15,16,17</sup> in which they carefully unravel an early claim of optical bistability<sup>18</sup>. It was initially found that hysteresis in the reflectivity was indeed observed at a nonlinear interface when pulsed light was used; this was initially interpreted to be an exhibition of bistability. However, the details of the measurements did not agree well with Kaplan's predictions<sup>12,13</sup>. Analysis of the situation and experiments with slower pulses revealed that bistability was unproven. Experiments with a medium having sufficiently large nonlinearity to be used cw then showed that there is also no real hysteresis, the observed apparent hysteresis being attributed to the short pulses used.

### Propagation in Nonlinear Waveguides

In waveguide devices having nonlinear bounding media, the same kinds of situations must be dealt with as those discussed above. Initial efforts at analysis of nonlinear guided waves have been made by Seaton et al<sup>19,20</sup>. In these papers, the waveguide itself is taken as linear with bounding media of nonlinear (Kerr) media. The nonlinear media are semi-infinite in extent. These papers provide the starting point for analysis of the devices discussed here.

The devices envisioned here have nonlinear media of finite thickness; no solution in simple functions can be obtained for this case. Instead, elliptic functions and integrals are obtained. Furthermore, our devices are channelized, i.e., they are bounded laterally as well. The crossing configuration means that spatial transients will be significant. One example: a wave propagating in the linear waveguide, upon encountering the nonlinear waveguide crossing it, will be partially reflected and, as well, some of the light will be lost into the substrate. This means that there will be set up, at least at the edge of the nonlinear medium, a standing-wave pattern of light. This may cause two phenomena. First, the standing wave pattern of intense light comprises a grating that converts the incoming light into reflected light or into light ejected into the substrate. So, it is possible that increasing the light intensity may merely lead to increased lost light rather than a transition into a transmitting state, as desired for the threshold unit. Second, the nonuniform intensity of light will lead to a nonuniform interaction with the material; such interactions are more difficult to analyze.

The preceding remarks indicate that analysis of the devices may involve new theoretical developments. This may be the case; however, there has been much work, especially in the Soviet Union (and referenced above) which lays the groundwork. Furthermore, the analyses made so far involve Kerr media, without regard to saturation. Real nonlinear media saturate with an index change that is only 1-3% of the low-intensity index. Hence, the more severe effects may not occur. There is good reason to expect that reasonable devices can be designed, analyzed and made; and that they will perform as expected. It will, of course, be important, in a world of technology accustomed to thinking in linear terms (additivity, superposition, decomposition, plane-wave analyses), to be especially careful to avoid both analytical and experimental paths that do not lead where linear thinking would presume.

## REFERENCES

1. C.M. Verber, R.P. Kenan, H.J. Caulfield, J.E. Ludman and P.D. Stilwell, Jr, "Pipelined Polynomial Processor Implemented with Integrated Optical Components", *Appl.Opt.* 23, 817-821 (1984).
2. Herwig Kogelnik, "Coupled Wave Theory for Thick Hologram Gratings", *Bell Syst.Tech.Jour.* 9, 2909-2947 (1969).
3. In such junctions, the addition of a  $\pi$ -phase shift into one input arm (using, for example, an electrooptic phase shifter) causes the output to be totally radiated into the substrate if there is light input to both input arms. If light is input into one arm, but not the other, or *vice versa*, half of the light is lost, but these are the only cases where light is transmitted into the output arm. Thus, the XOR truth table is implemented.
4. Colin T.Seaton, George I. Stegeman and H.G. Winful, "Intensity-Dependent Guided Wave Phenomena", in *Integrated Optical Circuit Engineering II*, Proceedings of the SPIE 578, 143-149 (1985).
5. P.W. Smith, I. Kaminow, P.J. Maloney and L.W. Stulz, "Self-Contained Integrated Bistable Optical Devices", *Appl. Phys. Lett.* 34, 62-65 (1979).
6. S.A. Akhmanov, A.P. Sukhorukov and R.V. Khokhlov, "Self-Focusing and Diffraction of Light in a Nonlinear Medium", *Soviet Physics Uspekhi* 93, 609-636 (1968).
7. B.B. Boiko, I.Z. Dzhilavdari and N.S. Petrov, "Reflection of Plane Light from a Nonlinear Transparent Isotropic Medium", translated from *Zhur. Prik.Spektros.* 23, 1511-1515 (1975).
8. N.N. Rozanov, "Pulsations fo the Radiation from a Laser with Frequency Dispersion", *Soviet Physics JETP* 36, 1074-1079 (1973). In this paper, the solution of the applicable nonlinear differential equations are displayed.
9. N.N. Rozanov, "Hybrid Conditions for Optical Reflection and Transmission by Nonlinear Media", *Sov.Tech.Phys.Lett.* 3, 239-241 (1977).
10. N.N. Rozanov, "Reflection of Plane Waves by Nonlinear Media", *Sov.Tech.Phys.Lett.* 4, 30-32 (1978).
11. N.N. Rozanov, "Nonlinear Reflections and Transmission of Limited Light Beams", *Opt.Spectrosc.(USSR)* 47, 335-337 (1979).



12. A.E. Kaplan, "Hysteresis Reflection and Refraction by a Nonlinear Boundary--a New Class of Effects in Nonlinear Optics", JETP Lett.24, 114-119 (1977).
13. A.E. Kaplan, "Theory of Hysteresis Reflection and Refraction of Light by a Boundary of a Nonlinear Medium", Sov.Phys.JETP45, 895-905 (1978).
14. Peter W. Smith, W.J. Tomlinson, Patrick J. Maloney and J.P. Hermann, "Experimental Studies of a Nonlinear Interface", IEEE J.Quantum Electron.QE-17, 340-348 (1981).
15. W.J. Tomlinson, J.P. Gordon, P.W. Smith and A.E. Kaplan, "Reflection of a Gaussian Beam at a Nonlinear Interface", Appl.Opt.21, 2041-2051 (1982).
16. Peter W. Smith and W.J. Tomlinson, "Nonlinear Optical Interfaces: Switching Behavior", IEEE J.Quantum Electron.QE-20, 30-36 (1984).
17. P.W. Smith and W.J. Tomlinson, "Nonlinear Optical Interfaces" in *Optical Bistability 2*, Charles M. Bowden, Hyatt M. Gibbs and Samuel L. McCall, ed., Plenum (1984), pp.364-374.
18. P.W. Smith, J.P. Hermann, W.J. Tomlinson and P.J. Maloney, "Optical Bistability at a Nonlinear Interface", appl.Phys,Lett.35, 846-848 (1979).
19. Colin T. Seaton, Jesus D. Valera, Richard L. Shoemaker, George I. Stegeman, John T. Chilwell and S.Desmond Smith, "Calculations of Nonlinear TE Waves Guided by Thin Dielectric Films Bounded by Nonlinear Media", IEEE J.Quantum Electron.QE-21, 774-782 (1985).
20. Colin T. Seaton, Jesus D. Valera, Bargro Svenson and George I. Stegeman, "Comparison of Solutions for TM-Polarized Nonlinear Guided Waves", Optics Lett.10, 149-150 (1985).

## APPENDIX

### Publication

S.C. Gustafson, J.A. Kirk, G.R. Little, R.P. Kenan and C.M. Verber, "Optical Implementation of Lumped Threshold Logic", SPIE Proceedings 564, San Diego, CA, August 18-23 (1985).

## OPTICAL IMPLEMENTATIONS OF LUMPED THRESHOLD LOGIC

S. C. Gustafson, J. A. Kirk, and G. R. Little

University of Dayton Research Institute  
300 College Park, Dayton, OH 45469

and

R. P. Kenan and C. M. Verber

Battelle Columbus Laboratories  
505 King Ave., Columbus, OH 43201

### Abstract

Optically implemented threshold logic systems that are characterized by thresholding operations concentrated at one functional location are considered. The objective is to identify architectures and associated integrated optical and holographic techniques that might be used to design superior register-level computation modules. A complete design for a lumped threshold 2-bit multiplier is presented as an example, and methods for general lumped threshold module synthesis are discussed.

### Introduction

Threshold logic has received attention recently because it may provide significant performance advantages for a broad range of mathematical operations and because it may have efficient optical implementations.<sup>1,2</sup> This paper considers lumped threshold logic systems, which are defined as systems in which thresholding operations are concentrated at one functional location. The objective is to identify architectures and associated holographic and integrated optical techniques that might be used to design register-level computation modules, such as pure-radix multipliers, multiply accumulators, etc., with commanding and enduring advantages in speed, power consumption, size, fault-tolerance, etc., over current and projected all-electronic alternatives.

Figure 1 shows two general types of systems that relate sets of inputs and outputs using weighting and thresholding operations. Here "weighting" refers to interconnects with selected connection strengths, and "thresholding" refers to decisions based on inequality criteria. Distributed threshold logic systems generally have numerous distinct elements of the same type, each of which performs weighting and thresholding functions. For example, each element could conceivably be an optical-input-optical-output multiple quantum well (MQW) gate, and all elements could be optically interconnected so that the thresholding operations would be distributed throughout the system. In contrast, lumped threshold logic systems generally have only two functional units, one for weighting and one for thresholding. For example, the weighting operation could be accomplished by passive or active (i.e., programmable) integrated optical diffracting elements, and the thresholding operation could be accomplished by photo-detectors at an optical-to-electronic output interface. In this case the (nonlinear) thresholding operation is global or concentrated at the output of the system.

### Lumped Threshold 2-Bit Multiplier

The 2-bit multiplier may be used as a simple example of lumped threshold logic design. Figure 2 is a 2-bit multiplier truth table for the multiplication of binary numbers  $x_1 x_0$  and  $y_1 y_0$  to obtain  $z_3 z_2 z_1 z_0$ . Suppose that the four input bits are represented by 0 if they are zero and by  $x_i = A_i \exp(i\phi_i)$ ,  $x_0 = A_2 \exp(i\phi_2)$ ,  $y_1 = A_3 \exp(i\phi_3)$ , and  $y_0 = A_4 \exp(i\phi_4)$  if they are ones. If these expressions correspond to waves in a geometrical optics approximation where all source-source, source-detector, and detector-detector distances are large compared to the wavelength, then the 2-bit multiplier may be designed as shown in Figure 3a. Here  $x_0$ ,  $x_1$ ,  $y_0$ , and  $y_1$  are optical point sources,  $z_0$ ,  $z_1$ ,  $z_2$ , and  $z_3$  are point photodetectors. The lines indicate optical paths, each of which may have a selected attenuation and phase shift that might be implemented by a hologram or integrated optical diffracting elements.

The required attenuations and phase shifts may be obtained by solving sets of simultaneous nonlinear inequalities derived from the truth table. For example, the 12th row and the  $z_2$  column of the table imply a signal at photodetector  $z_2$  that must equal or exceed a threshold  $T_2$ :

$$|A_1 \exp(i\phi_1) + A_3 \exp(i\phi_3) + A_4 \exp(i\phi_4)|^2 > T_2 \quad (1)$$

Similar expressions may be obtained so that each of the four output columns (labeled  $z_3$ ,  $z_2$ ,  $z_1$ , and  $z_0$ ) is described by a set of 16 (one for each table row) simultaneous nonlinear inequalities in 9 unknowns: four amplitudes ( $A_1$ ,  $A_2$ ,  $A_3$ , and  $A_4$ ); four phases ( $\phi_1$ ,  $\phi_2$ ,  $\phi_3$ ,  $\phi_4$ ); and one threshold ( $T_1$ ,  $T_2$ ,  $T_3$ , or  $T_4$ ). Solutions are to be found for each of these four overdetermined inequality sets (which involve terms such as  $A_1^2$ ,  $2A_1A_2\cos(\phi_1-\phi_2)$ , etc.) such that the amplitudes, phases, and thresholds obtained all have acceptable tolerances or ranges over which they may vary without affecting proper 2-bit multiplier operation.

One solution which involves only phase shifts (no attenuations) and which may have practical tolerances is given in Figure 3b, where the  $\phi$  column gives the phase shifts required for each of the four paths (in order) to each detector, the T column gives the threshold value for each detector when  $A_1 = A_2 = A_3 = A_4 = 1$ , and the  $\Delta T/R$  column gives the fraction of the total signal range on each detector over which its threshold may vary. Figure 4 is a histogram of  $\Delta T/R$  for output  $z_2$  generated by selecting each of the four phases for this output randomly from normal distributions with means at their design values and standard deviations equal to  $\arctan(.1)$ . These standard deviations correspond to 10% displacement of the phase vectors, and Figure 4 shows that such variations reduce the threshold tolerance  $\Delta T/R$  for output  $z_2$  from 37% to about 20%. Similar acceptable tolerances may be obtained for the other outputs and should be amenable to engineering design.

#### Optical Implementations

The 2-bit multiplier design described above is based on the ability of optics to provide noninterfering interconnections<sup>3</sup> which (1) are parallel in that interconnection time is essentially independent of interconnection length or weight and which (2) lead to system operation times essentially limited only by the response times of sources or detectors. These interconnections may be provided by passive diffracting elements in the form of an ordinary or bulk thin or thick film hologram in which light propagates approximately normal to the hologram plane. These interconnections may also be provided in integrated optical implementations by passive diffracting elements formed on or near a substrate surface such that light propagates approximately parallel to the surface. Such integrated optical implementations could use surface relief or photorefractive mechanisms to form the diffracting elements on GaAs, LiNbO<sub>3</sub>, glass or other substrates.

Integrated optics has potential for implementing lumped threshold computation modules with superior advantages in size, power consumption, reliability, etc. This technology also has potential for implementing real-time programmable interconnections or weightings using electro-optically modulated diffracting element structures (or, ultimately, all-optical nonlinear devices). This capability would be important, for example for neural network architectures that could perform "intelligent" adaptive and symbolic processing.<sup>4</sup> Figure 5a shows a direct implementation of programmable interconnections using a segmented array electro-optic grating.<sup>5</sup> Here the interconnection shown in Figure 3a are reordered so that no connection paths cross by providing a uniform optical input and applying one of two voltages to the grating segments in accordance with the input bits  $x_1, x_0, y_1$ , and  $y_0$ . Note that individual grating segments  $G_i$  are tilted at angles  $\theta_i$  so that parallel input beams  $\phi_i$  are directed to a detector with threshold T after weight  $W_i$  is applied as determined by programmable voltages  $V_i$ . However, since there must be a minimum deflection angle, the lateral separation D must become large as the number of segments N increases. Figure 5b shows one way of circumventing this problem using an integrated optical lens, which also permits (1) a common angle  $\phi$  for all grating segments and (2) the elimination at a stop (or monitoring) of undiffracted light. Figure 6 shows an integrated electro-optical channel-guide implementation. Here the channels are addressed through horns, and phase modulation is provided by surface electrodes. The output horns terminate in multimode regions whose outputs are plane waves which impinge on a surface grating at the Bragg angle. The contributions from the individual horns mix in the grating and produce an output which is a function of the phase differences between all pairs of input beams. Advantages of this arrangement are compactness and isolation of the detector from stray light.

Figure 7 shows how a hologram might be optically generated for implementing certain input-output relationships or truth tables (including the 2-bit multiplier truth table) in a lumped threshold system. One possibility, the Fourier transform hologram in Figure 7a, is multiply-recorded using object sources O related to output truth table elements and

reference sources R related to input truth table elements. Note that these sources need not be evenly spaced. The Fourier transform reconstruction in Figure 7b generates output truth table elements A given input truth table elements C. Using standard models of the holographic process it may be shown that the LxP output truth table matrix A is related to the NxP input truth table matrix C by

$$A = OR^*C \quad (2)$$

where O and R are LxM and NxM matrices describing the complex amplitudes used in recording the M-fold exposed hologram,  $m = 1, 2, \dots, M$ ,  $p = 1, 2, \dots, P$ , and  $+$  is the conjugate transpose operation. An important aspect of Eq. (2) is that although many exposures may be used to record the hologram, the ability of the hologram to represent input-output relationships is described by no more than the NL complex elements of  $OR^*$ . In the 2-bit multiplier, for example, where  $N = L = 4$  and  $P = 16$ , only 16 complex parameters are available to relate 64 input bits to 64 output bits. This suggests that not all possible truth tables are realizable in an optically recorded hologram of the type considered here. An analogous situation is that not all logic functions can be implemented by single threshold logic elements.<sup>6</sup>

It would be useful to at least approximately solve Eq. (2) for  $OR^*$  in terms of C and A. This matrix equation is generally over determined, and least-squares or pseudoinverse methods might be used to obtain an approximate solution. The (row by row) least-squares solution, for example, is

$$OR^* = AC^+(CC^+)^{-1} \quad (3)$$

While this solution may not yield the desired truth table realization in a lumped threshold system, it may serve as a starting point for a steepest descent or other computer search for desired solutions. Such solutions should maintain the desired input-output relationship when the matrix elements are varied over an acceptable tolerance range. The geometrical optics phase-only solution for the lumped threshold 2-bit multiplier described in Figure 3b is such a solution and may be used to derive an  $OR^*$  matrix in which all elements have unit magnitude:

$$OR^* = \begin{bmatrix} 1 & 1 & 1 & 1 \\ (-\sqrt{15} + 1)/4 & (-1 + \sqrt{31})/2 & (\sqrt{15} + 1)/4 & (-1 - \sqrt{31})/2 \\ 1 & 1 & -1 & 1 \end{bmatrix} \quad (4)$$

A particular implementation of this solution for holographic recording is  $O =$  the 4 x 4 identity matrix and  $R = (OR^*)^+$ . Note that although the above analysis implies three-dimensional holographic systems, integrated optical assemblies of diffracting elements similar in function to bulk holograms may be practical. This possibility is related to the observation that the multiple truth table "images" to be recorded and reconstructed, although often highly cross-correlated, may be relatively simple or low-resolution bright-spot-dark-spot patterns.

#### General Lumped Threshold Module Synthesis

Optical or computer generated hologram synthesis of the weighting or interconnecting units required for lumped threshold computation modules will generally require knowledge of the amplitude and phase patterns on the hologram that yield the correct truth table input-output behavior with maximum weight and threshold tolerances. In the case of the geometrical optics two-bit multiplier design of Figure 3, expressions governing input-output behavior were easily obtained. This favorable situation may be uncommon in the design of the generally smaller, more efficient, etc., lumped threshold modules for which geometrical optics approximations do not apply.

Consider, for example, the derivation of eight far-field holograms, each with the same two design parameters, that implement, in a lumped threshold system, the eight positive-threshold two-Boolean variable functions (i.e., the eight out of the sixteen functions for which two zero inputs yield a zero output). Figure 8 shows a simple format consisting of a screen with two pinholes separated by a distance y. One pinhole is covered by a phase-shifting film  $\theta$ ; there is a detector d and lower and upper mutually coherent point sources l and u. In the far-field approximation the distances b and y and the wavelength  $\lambda = 2\pi/k$  must be small compared to the distance s. With this approximation and with b fixed, the problem reduces to finding values of y and  $\theta$  such that the detected signals  $I_i$  for only source l on,  $I_u$  for only source u on, and  $I_b$  for both sources on have all six possible

inequality relationships. Referring to Figure 8 for definitions, the following approximate expressions may be derived:

$$\begin{aligned}
 A_f &= \exp i[k(w + s)] + \exp i[k(v + r) + \theta] \\
 A_u &= \exp i[k(w + s)] + \exp i[k(u + r) + \theta] \\
 I_f &= |A_f|^2 = (ks)^2(x^2 + ax)^2 + 2\eta(ks)(x^2 + ax) + \eta^2 \quad (5) \\
 I_u &= |A_u|^2 = (ks)^2(x^2 - ax)^2 + 2\eta(ks)(x^2 + ax) + \eta^2 \\
 I_b &= |A_f + A_u|^2 = 2(ks)^2[(x^2 + ax)^2 + (x^2 - ax)^2] \\
 &\quad + 8\eta(ks)x^2 + 4\eta^2 - 4(ks)^2(ax)^2
 \end{aligned}$$

where  $x = y/s$ ,  $a = b/s$ , and  $\eta = \theta - \pi \approx 0$ . Figure 9 is a graph of the approximate expressions for  $I_f$ ,  $I_u$ , and  $I_b$  versus  $x$  for  $\lambda = 628 \text{ nm}$ ,  $b = 10 \text{ }\mu\text{m}$ ,  $s = 10 \text{ cm}$ , and  $\eta = .004$ . Note that four of the six inequality relationships can be satisfied using the plotted values; the other two relationships can be satisfied for other values of  $\eta$ .

The example of Figure 8 and Eqs. (5) indicates the possible complexity of general (physical optics) lumped threshold module synthesis. Greater complexity may be anticipated if Fresnel rather than Fraunhofer diffraction conditions are allowed and if the input-output truth tables are large. One approach to such synthesis problems is to perform additional post-photodetection processing and to employ logical reduction and residue arithmetic techniques to reduce the effective size of the truth tables to be realized. The approach considered here seeks alternatives to requirements for conversion into and out of residue arithmetic and for additional all-electronic processing.

Further work on lumped threshold logic should emphasize studies of the types and sizes of realizable truth tables and should seek general methods for synthesizing holograms such that the required truth tables are realized with acceptable threshold and other tolerances. A straightforward but perhaps limited approach to these studies is to investigate the number of holograms with specified resolution and cross correlation characteristics that can be multiplexed on a single recording medium. A more general approach may be to obtain expressions - generally large sets of overdetermined simultaneous nonlinear inequalities - that fully describe a desired optically implemented lumped threshold module and to find optimal solutions for them using nonlinear programming techniques. This approach will probably require the use of supercomputer facilities, but in many cases it may be the best approach for obtaining designs with optimum performance characteristics.

#### References

1. R. Arrathoon and M. H. Hassoun, "Optical Threshold Elements for Digital Computation," Opt. Lett. 10: 143-145 (1984).
2. S. C. Gustafson, D. L. Flannery, and R. O. Winder, "Analysis of Threshold Logic for Application to Optical Computing," Proc. SPIE 564, no. 24, (1985).
3. J. W. Goodman, F. J. Loonberger, S. Y. Kung, and R. A. Athale, "Optical Interconnections for VLSI Systems," Proc. IEEE 72: 850-866 (1984).
4. D. L. Flannery, S. C. Gustafson, and L. M. Vail, "The Use of Multiple Associative Memories for Symbolic Processing," White paper, 15 pp., (9 Aug. 1985).
5. C. M. Verber, "Integrated Optical Approaches to Numerical Optical Processing," Proc. IEEE 72: 942-953 (1984).
6. P. M. Lewis and C. L. Coates, Threshold Logic, Wiley, NY, pp. 458-467, (1967).
7. T. K. Gaylord, M. M. Mirsalehi, and C. C. Guest, "Optical Digital Truth Table Look-up Processing," Opt. Eng. 24: 48-58 (1985).

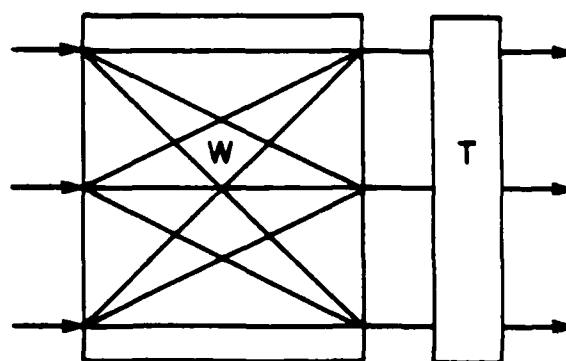
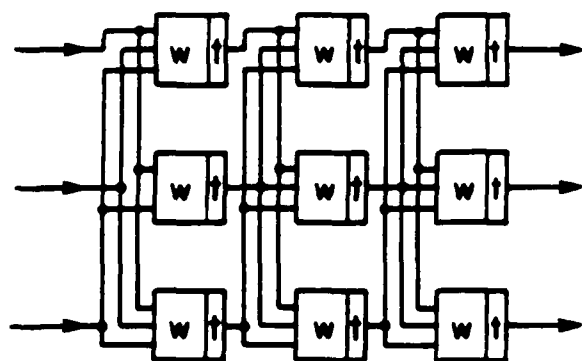


Figure 1. (a) Distributed threshold logic, (b) lumped threshold logic.

$x_1$	$x_0$	$y_1$	$y_0$	$z_3$	$z_2$	$z_1$	$z_0$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Figure 2. 2-bit multiplier truth table. The 12th row of the input and of the  $z_2$  column are boxed.

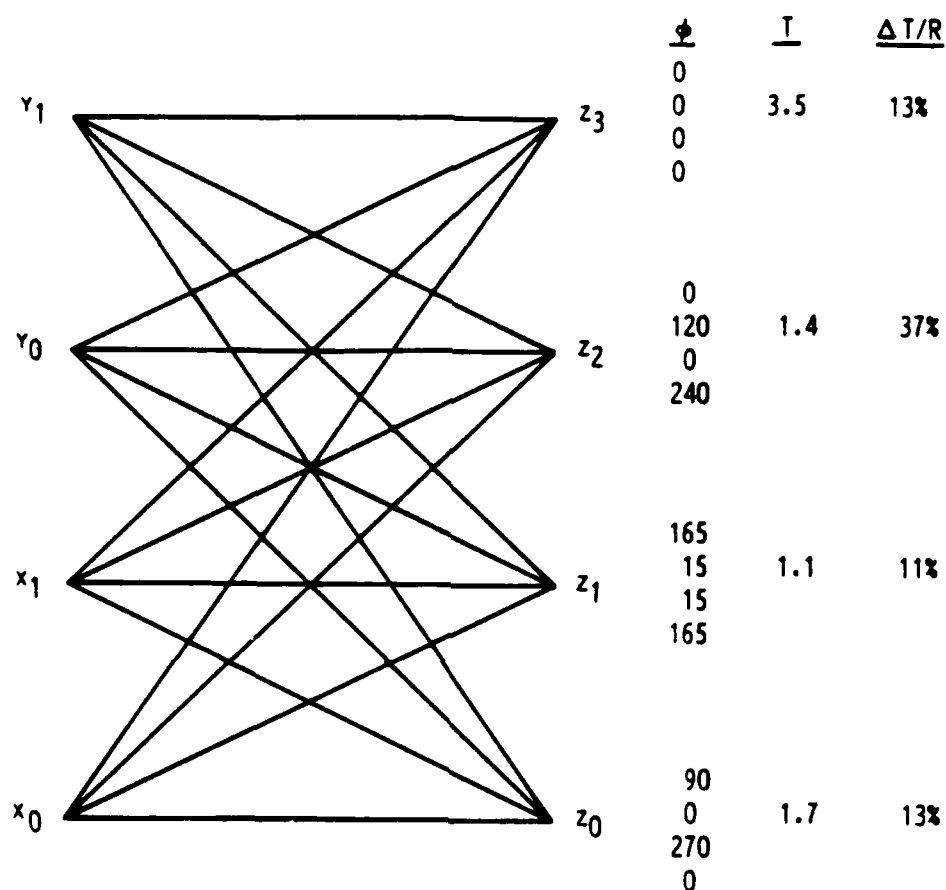


Figure 3. Lumped threshold 2-bit multiplier. (a) Interconnections from sources  $x$  to detectors  $z$ , (b) no-attenuation solution for interconnection phases  $\phi$ , detection thresholds  $T$ , and threshold tolerances  $\Delta T/R$ .



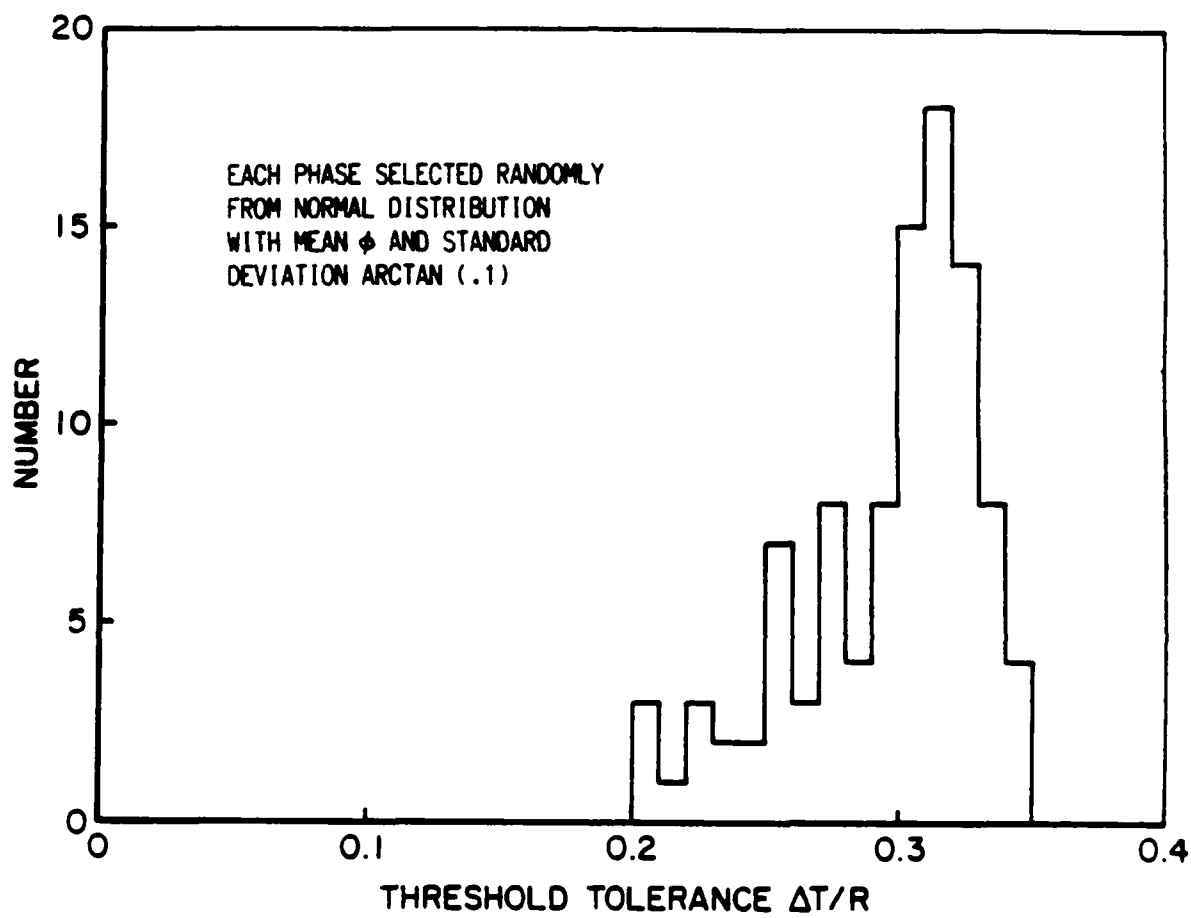


Figure 4. Histogram of threshold tolerance for output  $z_2$ .

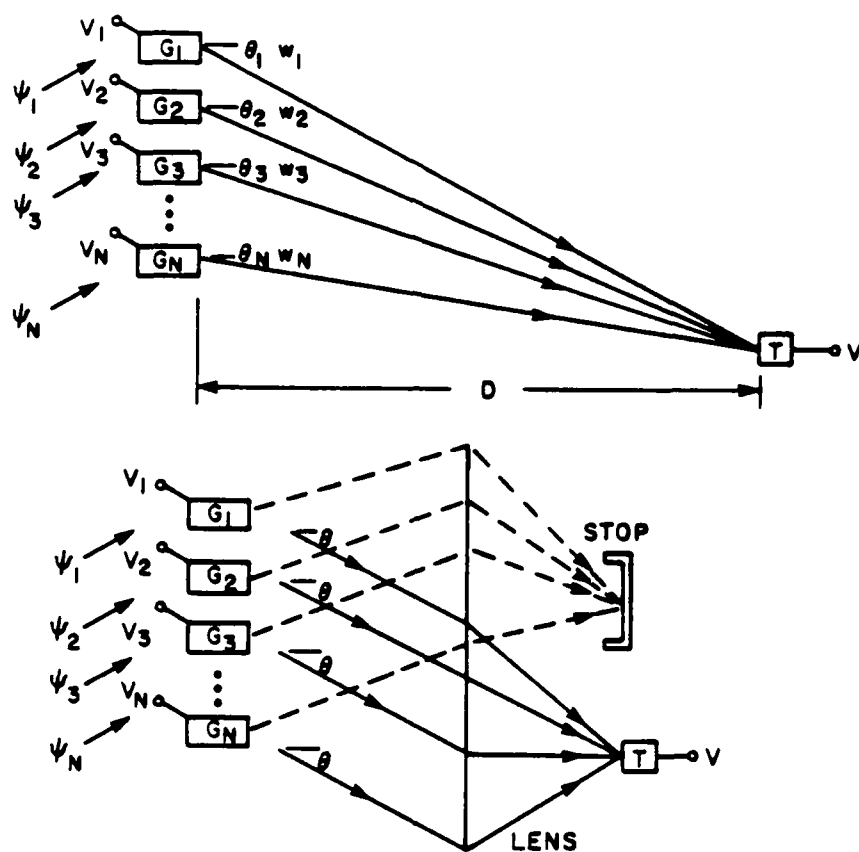


Figure 5. Integrated electro-optic grating arrays for programmable threshold logic. (a) Direct implementation, (b) arrays with lens.

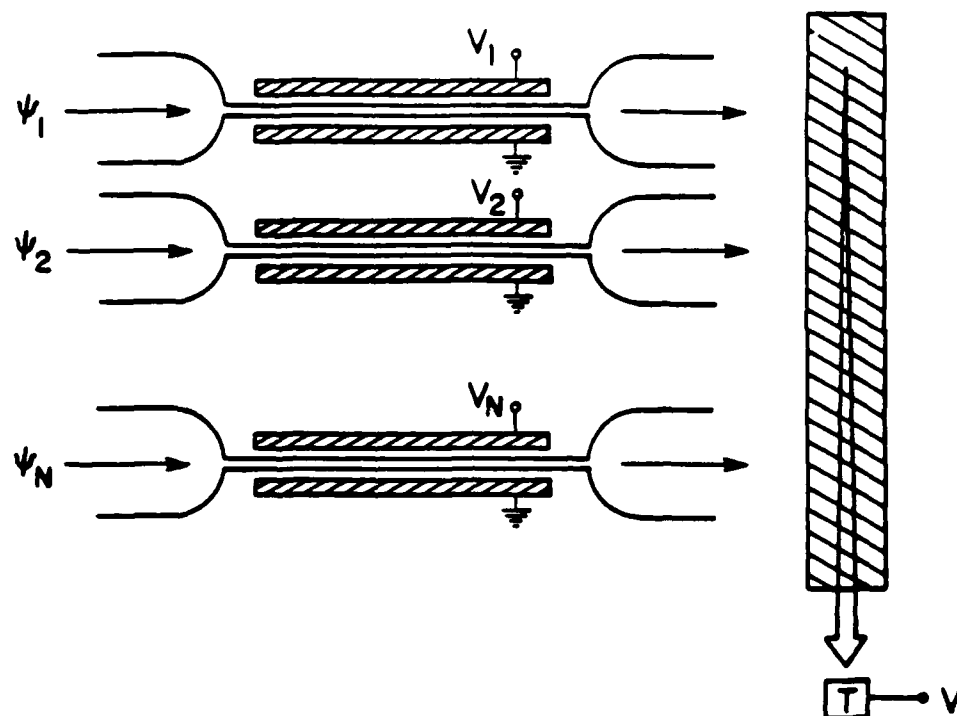


Figure 6. Integrated electro-optic channel arrays for programmable threshold logic.

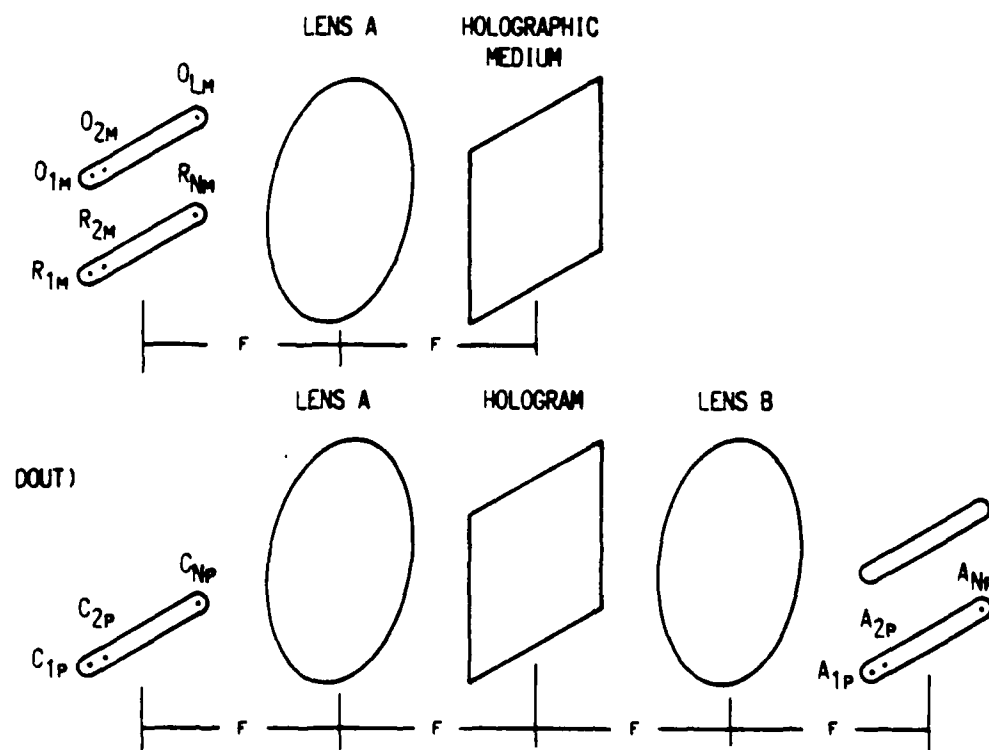


Figure 7. Holographic synthesis of lumped threshold logic. (a) Recording, (b) reconstruction (truth table readout).

[illegible]

Figure 1 is a plot showing the relationship between  $10^4 I_l, I_u, I_b$  (Y-axis) and  $10^4 x$  (X-axis). The Y-axis ranges from -1 to 10, and the X-axis ranges from -3 to 2. Three curves are plotted, labeled  $I_l$ ,  $I_u$ , and  $I_b$ . The curves show a minimum value around  $10^4 x = 0$  for  $I_l$  and  $I_u$ , and a minimum value around  $10^4 x = -1.2$  for  $I_b$ .

8-27

**THEORETICAL INVESTIGATION OF THE EFFECTIVE UTILIZATION OF  
PARALLELISM AND CONNECTIVITY IN OPTICAL COMPUTERS**

Demetri Psaltis and Yaser S. Abu-Mostafa, Principal Investigators

CALIFORNIA INSTITUTE OF TECHNOLOGY

CALIFORNIA INSTITUTE OF TECHNOLOGY

Principal Investigators:

Demetri Psaltis  
Yaser S. Abu-Mostafa

Theoretical Investigation on The Effective Utilization of  
Parallelism and Connectivity in Optical Computers.

Abstract

Parallelism and global connections are the main features of optical computers. The computation power of these features is quantified using a new measure based on the degrees of freedom that are available. This measure of computation power is then related to the versatility of the computer and the complexity level of the problems it can tackle. The practical significance is demonstrated by optical implementation of FPLA's and associative memories. The quantitative measure and the methodology are equally applicable to general non-optical computing systems.

## I. Introduction

In contrast with the digital world, it is much easier to achieve global communication than to implement sophisticated local computation in optical systems. In fact, a tremendous level of parallelism can be achieved with the simplest form of optical systems. The power of parallelism and global communication is therefore the essential merit for building optical computers, and it is essential to assess this power in quantitative terms. It may seem at first glance that a measure of the computation power of global interconnections without regard to what kind of computing elements are being interconnected has no real significance, since the situation is different if for example we are connecting Cray computers or AND gates. However, if we are able to assess the "contribution" of interconnections to the computation power of the whole system, we will have a useful measure of their role. In this paper, we define such a measure based on the computational degrees of freedom, apply it to optical systems to assess their power, interpret this power in terms of optical implementation of FPLA's and associative memories, and argue that the measure is not restricted to optics but is equally valid in a general computing system.

In section II, we familiarize the reader with optical computers and hybrid systems of optical-digital computers. In section III, we discuss computation power briefly and quote some results relating the versatility of a computing system to its ability to tackle problems of a certain level of complexity. The measure of computation power is introduced and discussed in section IV, and then applied to optical computers. In section V, two classes of examples are discussed to show how the measure of computation power is directly reflected in practical situations.

## II. Optical Computers

A schematic diagram of a generalized optical information processing system is shown in Figure 1. It consists of two planes which are optically interconnected. The interconnection pattern is determined by the optical system that is placed in the intervening space. If optical feedback is used the two planes can be physically merged into one and two way communication is possible. Analog signal processing can be performed with the general structure of Figure 1 by assigning analog weights to each interconnection between individual pixels at the input and output planes. Accumulation of the weighted samples at the output by either spatial or temporal integration results in the implementation of linear transformations of the data placed at the input plane. The implementation of nonlinear optical computers has also been proposed [1,2,3] using the same basic structure shown in Figure 1. In such systems, nonlinear

computations are performed at the two planes by either optoelectronic semiconductor circuits [3] or nonlinear optical components [1,2]. In either case, the basic structure of the optical computer is very similar to an analog signal processor, however there is a very important shift in emphasis and outlook. In an optical computer, the elementary computations are being performed at the two planes and the primary purpose of the optical system is to provide global communication among the computing elements. This is obviously true in the case of optical interconnection of electronic processing components [3], but it is equally true if optical logic elements [4] are used, since little or no advantage over electronics can be derived by configuring an optical logic circuit in a planar geometry.

An optical computer in principle combines the best of two worlds. Planar technology (VLSI or optical) is used to perform non-linear logic and the third dimension is used to provide optical communication between the computing elements in the plane. Indeed, the primary motivation for the proposed development of optical computers has been the solution of existing and anticipated communication bottlenecks in VLSI [3,1]. The ability to configure the interconnection pattern in the third dimension is a unique property of optics and it does not only

provide an efficient means for parallel communication among computing elements that are well separated in the plane, but also it makes global communication among all the computing elements in principle possible. In an optical computer of this type, the computing elements that are being interconnected perform relatively simple computations since only relatively local interconnections are possible with a planar technology. The assessment of the potential power of optical computers, reduces then to an assessment of the relative importance of the sophistication of the computing elements versus the communication capability among the elements. For illustration purposes we present two hypothetical curves in Figures 2a and 2b. The vertical axis is the computational power of a system consisting of  $N$  parallel computing elements, plotted as a function of the number of interconnections among the elements. Several curves are drawn in each diagram, corresponding to different computational power of the individual computing elements. Figure 2a is drawn under the hypothesis that the relative importance of the communication capability is dominant; as the number of interconnections is increased, the computational power of the system rises and for large, fully interconnected networks the computing power of the individual elements only marginally affects the overall system for large  $N$ . The opposite situation is depicted in Figure 2b. Interconnecting a large number of simple elements does not result in a powerful system and a locally interconnected network of very powerful computers is essentially the best that can be done. If in reality the



situation is as shown in Figure 2a, then we can be very optimistic about the prospects of optical computers. In this paper, we express our conviction that indeed the situation is as depicted in Figure 2a and present quantitative arguments in support of this conjecture.

### III. Computation Power

Computation power (the variable plotted in Figure 2) is a quantity that must encompass two aspects. The first is what we call raw computing power: the maximum number of elementary operations that can be performed per unit time. The main factors affecting the raw computing power are technological (switching speed for instance), and parallelism. On the other hand, a meaningful measure of computation power must include the ability of the computer to handle complex problems, and it is important to make the distinction between large problems and complex problems. Forming the inner product between two large vectors, for example, is a problem of very low computational complexity since only one operation per vector element is required. Clearly parallelism can help solve this problem faster since products between the individual elements of the vectors can be separately formed; global interconnections are not essential since the

products can be added pairwise. How parallelism can be help solve a problem of inherent high complexity is not equally clear. A complex problem has the property that local decisions cannot be made until essential information has been communicated from basically the entire input data [5]. Thus useful computation can progress only when all the input information has been considered by the individual elements. For a parallel processor, this implies that all partial results need to be globally communicated. It is this notion that forms the basis for our conviction that communication capability becomes the dominant factor in determining the computation power of a highly parallel computing structure, rather than the capabilities of the local computing elements.

Complex problems of this type arise in disciplines such as pattern recognition, and are characterized by the lack of a regular structure that would admit a short, systematic algorithm for solution. Instead, the problem has to be broken into a very large number of basically unrelated cases that must be considered in order to solve the problem. A convenient measure of the complexity is the logarithm of the number of cases, namely the entropy  $H$  of the problem [5]. The entropy of the problem is clearly related to the size of the system that can handle it, since the system must keep track of all the cases and the number of these cases is exponential in the entropy. Indeed, a direct relation between the entropy  $H$  and the cost  $C$  of the system is derived in [5]. On the other hand, the number of problems that

can be handled by a system, by proper programming, is also directly related to its size. For example, if the memory capacity is large, the number of programs and look-up tables that can be stored is also large, and hence the number of different problems that can be tackled. Through these two relations, we can link the number of problems that can be tackled to the level of complexity (or entropy) that can be handled [5]. In other words, a computing structure that solves a certain class of complex problems also solves a large number of different problems, and vice versa.

When we interpret these relations in terms of optical computers, a definition of "size" or "cost" is needed. On the one hand, the definition should be independent of the particular technology in question, i.e., it should be equally applicable to any other technology that yields computing systems. On the other hand, it should have direct practical relevance to the ability of the system to do computation. In what follows we introduce the number of degrees of freedom in a computing circuit as the appropriate measure for computation power and assess the computation power that is associated with optical interconnections based on this definition. The examples that follow will demonstrate the computation power associated with the degrees of freedom in terms of the number of problems as well as the complexity of problems that can be handled.

#### IV. Degrees of Freedom

Whereas the speed of computation and the size of the problems that can be handled by a device are indeed important factors in its computation power, the versatility of the device becomes the essential factor of computation power when we address general-purpose computation. If a device does one very complicated computation task very quickly, it could still be very hard to embed an arbitrary computation problem in this device. For example, the Fourier transform and linear filtering of two-dimensional functions come very naturally in optical systems. These are not trivial operations, they are considered more difficult than simple logic operations for example. However, it is not as easy to implement a wide class of these simple logic operations in optical systems.

We are concerned here with quantifying the versatility of a computation device as a measure of its computation power. After defining this measure, we apply it to optical connections and hence measure the power of this single most important computational feature of optical systems. Once we arrive at this quantitative measure, we demonstrate that we can get as much computation from optical connections as we should expect from a general device that has the same value of this measure of computation power.

A computation device can be "programmed" to be in one out of a number of possible states. After programming the device, it is in the state for solving a specific problem. The number of possible states of a device is a measure of the number of different problems it can possibly handle. For example, a Field Programmable Logic Array (FPLA) can be programmed to simulate any of a large number of Boolean functions. To program it for a certain function, we preset a number of internal parameters thus defining which function we are simulating. The number of parameters under our control determines the number of ways we can program the chip, hence the number of functions we can simulate. These parameters constitute degrees of freedom for computation versatility.

Given a device to be used as a component in a computation system, we define the computation power  $c$  of that device to be the number of binary parameters that can be set independently to fix the characteristics of the device, i.e., the degrees of freedom. In the general case of non-binary or dependent parameters, the measure is given by the logarithm to base 2 of the number of different ways in which the device characteristics can be fixed. For example, a Programmable Read-Only Memory (PROM) with  $n$  address lines and one data line has  $2^n$  memory locations each of which can contain 1 or 0. Each pattern of 1's and 0's corresponds to a distinct Boolean function when the address lines are considered as the input Boolean variables and the data line is considered as the output Boolean function. Therefore, the computation power  $c$  of this PROM is  $2^n$ . This power is reflected by simulating a large number of functions as well as certain functions of high complexity. It is quite general that the ability to solve complex problems is associated with the ability to solve a large number of problems [5].

A general device having  $c = N$  can be thought of as having  $N$  "cells" which we can load with either 0 or 1 thus fixing the characteristics of the device in a unique manner. How does this definition apply to optical connections? Consider two planes each containing  $M \times M$  pixels (Figure 1), and a hologram that determines which pixels of the first plane are connected to which pixels of the second plane. Each connection can be either present or absent independently of the others, and there are  $M^2 \times M^2$  such connections possible. There, the computation power of optical connections between these two planes is given by  $c = M^4$ . In the case of weighted connections, the computational impact of the weights can be incorporated in the planar logic operations to be performed. This impact has been found experimentally to be limited [9], which is expected from results in threshold logic which show this to be true unless the technology can accommodate an exponential dynamic range of weights [7].

For  $M = 512$ , the value of  $c$  is  $2^{36}$ . To appreciate this value, let us compare it to a PROM with 64K bytes of memory. In this case, we have  $2^{16} \times B$  degrees of freedom, hence  $c = 2^{19}$ . The optical connections are therefore as powerful as approximately 130,000 such PROMs. It is clear that using these PROM's, one can implement some functions of very high complexity. Hence, the hologram should be able to play a computational role of the same sophistication. We now demonstrate this role by example.

## V. Examples

It is clear that  $c$  imposes an upper bound on the versatility of a device, since we cannot do more than what the degrees of freedom allow us to do. However, it is yet to be determined whether we can put these degrees of freedom to work for us in a general computation task. In this section, we address the usefulness of optical connections as we incorporate them in a system designed to solve a large class of problems. We shall discuss the optical implementation of two different examples of fairly general structure and computational usefulness, in which optical connections play the central role. These are Boolean functions and associative memories.

Consider a Boolean function which has a sum-of-products expansion (Boolean expression formed by ORing terms, each being an ANDing of Boolean variables or their negations [B]) with a relatively small number of terms. This is a variation of low-entropy functions [5] which include most pattern recognition decisions. For example, we fix a large number of independent Boolean variables, say  $n$ , and consider those functions of  $n$  variables which can be expressed as the sum of at most  $2^{18}$  product terms. Using optical connections, we can implement any such function by varying only the hologram within an otherwise simple fixed architecture. In other words, all the degrees of freedom come from optical connections.

The architecture follows Figure 1. The first plane will have  $3n$  pixels of unit intensity.  $n$  of these correspond to the  $n$  variables and will be on if the variables assume the value 1, another  $n$  pixels correspond to the negations of the variables, while the extra  $n$  pixels will be always on. The second plane consists of a  $512 \times 512$  threshold elements with fixed threshold at  $n - 1/2$  units of intensity. The simulated function will have the value 1 if one or more threshold elements are excited. The hologram is designed to simulate the given function by assigning a product term to each threshold element in the second plane. Each literal (variable or negated variable) appearing in each product term is connected to the corresponding threshold element, and if the number of literals in the product term is less than  $n$ ,

some always-on pixels are connected to the element to make the total number of connections equal to  $n$ .

This implementation may not be the most efficient, but it illustrates exactly the title of this article. Without the hologram, the system is a simple repetitive structure with no problem dependency and thus no capability of doing any real computation. The optical connections simulated by the hologram fully capture the computation aspect. A simple enumeration argument will show that a huge number of degrees of freedom is indeed required to simulate these functions. This example, and more complicated versions thereof, demonstrate the utilization of the degrees of freedom of optical connections.

Another example that demonstrates the computing power afforded by the degrees of freedom of optical interconnects in the implementation of the nearest neighbor search operation according to the model that was described by Hopfield [6] for neural networks. The basic architecture is again as in Figure 1 except that feedback is used from the output back to the input [9]. The computing elements at the output plane are threshold elements, one at each pixel. An interconnection between the  $i$ th pixel at the input and the  $j$ th pixel at the output is made if  $\sum_m v_{im} \cdot v_{jm} > 0$ , where  $v_{im}$  are binary words,  $n$  bits long each,

$m$  that are stored in the system. When the initial state of the system is set according to an external stimulus, the state of the system generally converges to the stored binary vector that is at the shortest Hamming distance from the initial state. Thus the system performs a nearest neighbor search, a fundamental operation for pattern recognition, associative memories and error correction. It is possible to implement this model with electronic components, using locally interconnected, pipelined (systolic) multipliers/ accumulators for simulating the interconnection matrix and an array of thresholding elements. Thus the global communications are not essential. However, the electronic implementation requires  $n^2$  locally connected elements in order to produce a product vector in each cycle, whereas the optical implementation requires only  $n$  very simple computing elements and  $n^2$  communication links. The overall number of degrees of freedom is the same in both cases, demonstrating that each optical connection in this example contributes the same computing power to the overall system as an individual computing element.

In general, the dramatic increase in the degrees of freedom that are created by optically interconnecting a large number of computing elements in a planar structure can, in principle, be translated to a proportional increase in overall computing power. Two examples were given that exemplify specific methods for

tapping this potential, but the challenge still remains to develop algorithms that are appropriate for computing structures of this type and solve technological problems that will make large networks of optically connected computing elements feasible.

### References

- [1] A. Huang, "Architectural considerations involved in the design of an optical digital computer, "Proceedings of the IEEE, Vol. 72, pp. 780-786, 1984.
- [2] A. A. Sawchuck & T. C. Strand, Digital optical computing, "Proceedings of the IEEE, Vol. 72, pp. 758-779, 1984.
- [3] J. W. Goodman et al., "Optical interconnections for VLSI systems," Proceedings of the IEEE, Vol. 72, pp. 850-866, 1984.
- [4] H. M. Gibbs et. al., "Optical bistable devices: The basic components of all-optical systems?," Opt. Eng., Vol. 19, pp. 463-468, 1980.
- [5] Y. Abu-Mostafa, Complexity of Information Extraction, Ph.D. Thesis, Caltech, 1983.
- [6] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," Proc. Natl. Acad. Sci. USA, Vol. 79, pp. 2554-2558, 1982.
- [7] P. M. Lewis & C. L. Coates, Threshold Logic, Wiley, 1967.
- [8] Z. Kohavi, Switching and finite automata theory, McGraw-Hill, 1978.
- [9] D. Psaltis & N. Farhat, "Optical information processing based on an associative memory model of neural nets with thresholding and feedback," to appear.

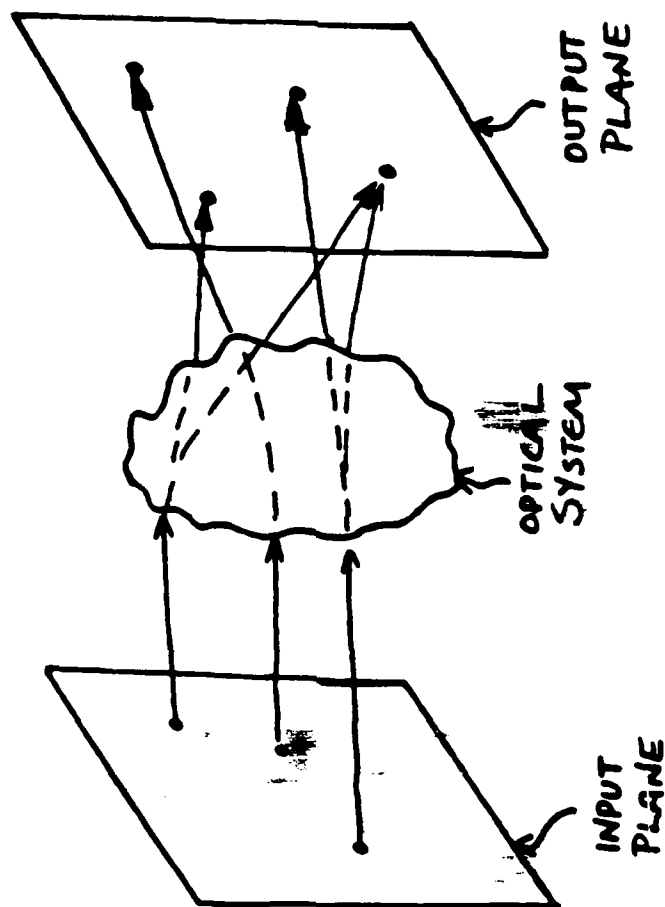


Figure 1. Optical Information Processing System

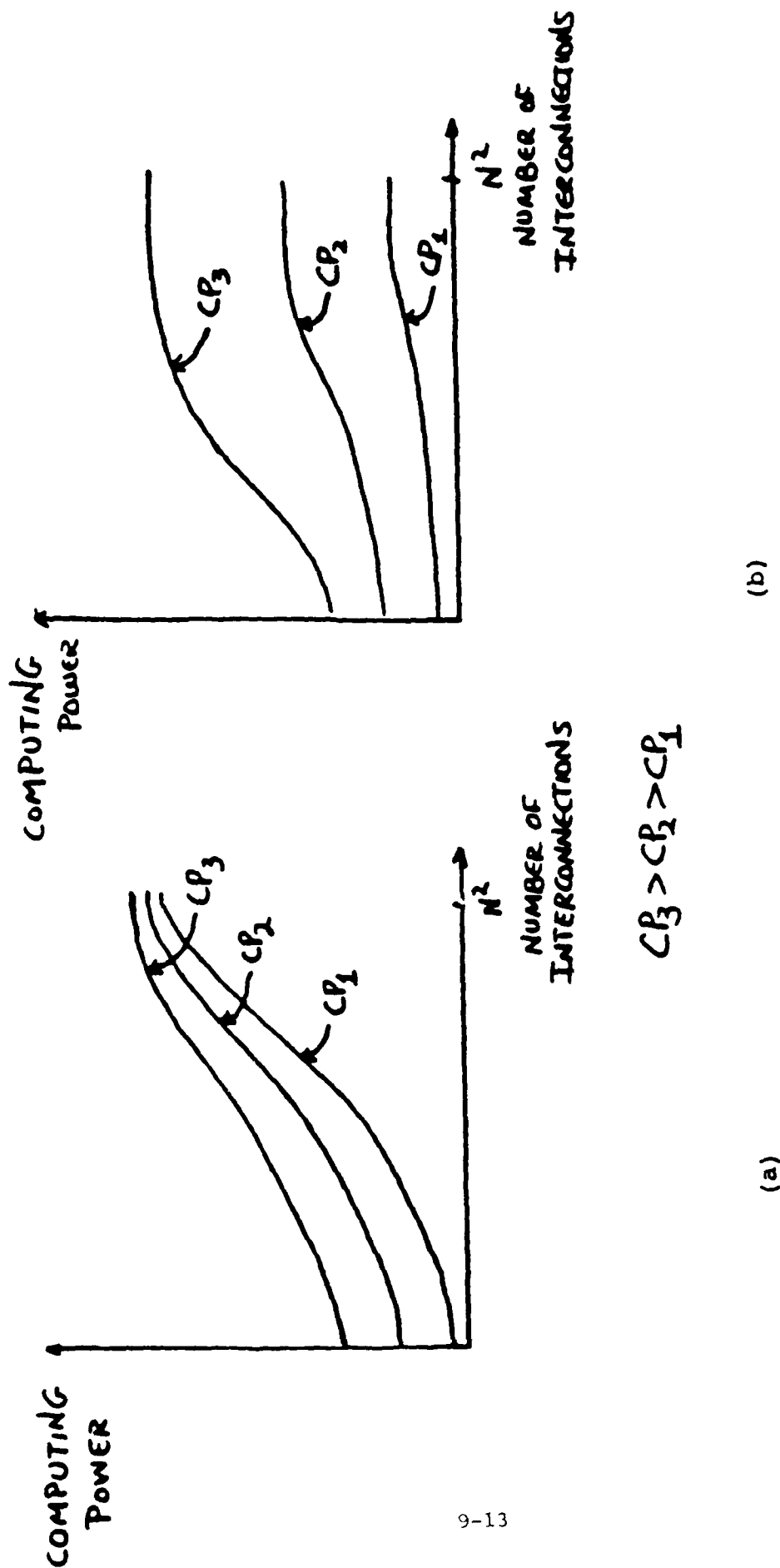


Figure 2: Computation Power of  $N$  parallel Computing elements, for large  $N$ , as a function of the number of interconnections between the Elements (hypothetical plots)

- (a) The case where global communication dominates local computation.
- (b) The case where individual computing power is a dominant factor.



**2-D NEIGHBORHOOD PROCESSOR**

Roger S. Putnam and H. J. Caulfield, Principal Investigators

**AERODYNE RESEARCH, INC.**

AERODYNE RESEARCH, INC.  
45 MANNING ROAD  
BILLERICA, MA. 01821

PRINCIPAL INVESTIGATORS:

DR. ROGER S. PUTNAM  
DR. H. J. CAULFIELD

"2-D NEIGHBORHOOD PROCESSOR"

ABSTRACT

The research identified a bottleneck in electronic image processing which is the large number of multiplications and additions required, and proposed solutions using optical parallel multiplication and addition. One technique utilized a film matrix multiplier and an array of LED's, and the second system employed a single channel acoustooptic cell and demonstrates near-optimal two dimensional data flow. A technique for assessing the advantages and disadvantages of optical processors is also presented, with examples such as an equally fast and more accurate electronic implementation of an optical matrix-vector multiplier which accomplishes an entire fourier transform in parallel.

## TECHNICAL SUMMARY

### 1. Objectives

The objective of the research was to investigate optical systems capable of performing two dimensional neighborhood operations. The objective arises due to the increased need for image processing and pattern recognition as applied to vision systems for robotics, autonomous vehicles, reading machines, and especially the SDI surveillance requirements which would completely overwhelm present electronic techniques.

### 2. Description of Work Performed and Results

The work was accomplished in two cycles of analysis and invention. During the first cycle an analysis of the weaknesses present in electronic image processors was made, followed by the invention of an optical technique which accomplished the same general task but with a higher degree of parallelism. The device included LED's to represent image pixel intensities and a film mask used as the multiplying filter weights.

The second cycle in the research involved analysis of the new LED-Mask device including a serious comparison with an electronic analog of the optical system. Another optical device described in the literature was similarly analyzed by comparing it to an electronic analog invented for the sole purpose of properly assessing the actual advantages and disadvantages inherent in using optics in the design. In both cases the electronic analogs of the optical devices were found to be superior to the optical system in several aspects, equivalent on some issues and inferior on only one point.

The last piece of the research was the invention of a 2-D Convolution Filter using an acousto-optic cell which accomplishes several of the objectives in speed, data flow efficiency, accuracy, programmability and simplicity. The area of a 2-D filter that is implemented can also be expanded without affecting the speed, and with small changes in the data flow efficiency. This acoustooptic cell device requires a minimum of additional components including a single diode laser, one lens, two detectors and a film mask.

### 3. Conclusions and Recommendations

The work was accomplished in cycles of analysis and invention which facilitated unveiling the true advantages and disadvantages of several optical signal processing systems. The optical devices that we proposed were also subjected to the same analysis. The result is that we have presented an analysis technique for future optical processing schemes, as well as a useful optical system that implements a large area two dimensional convolution filter.

The need for efficient image processing will continue to increase as is evidenced by the almost nonexistence of visual senses in robotics, by the recent initiation of reading machines and initial tests for autonomous vehicles, and by the vast data requirements for artificial intelligence applications such as missile tracking and identification. We encourage more research leading to generic 2-D signal processing because it is, in fact, now a bottleneck in several fields.

We also recommend evaluating the advantages and disadvantages of optical processors by inventing an electronic analog of the system, which will assist in identifying which of the features of the optical system are in fact specifically due to the optics. Further we recommend that familiarity with digital electronic implementations will encourage a cross breeding of ideas with the analog electronic and optical processing systems.

AD-A169 882

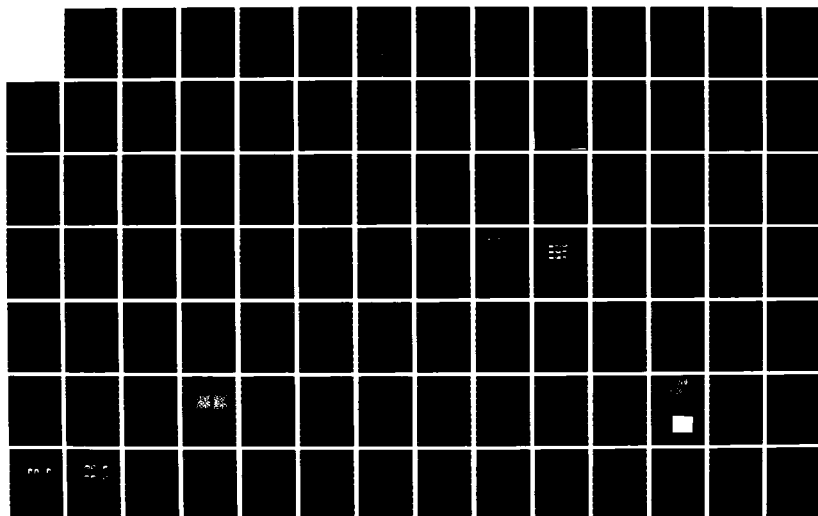
OPTICAL PROCESSING(U) DAYTON UNIV OH RESEARCH INST  
S C GUSTAFSON 31 DEC 85 UDR-TR-85-148 N00014-85-K-0479

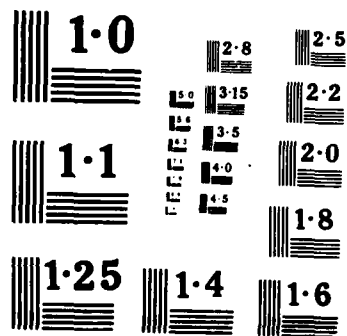
5/6

UNCLASSIFIED

F/G 9/2

NL





## APPLICATIONS OF THE OPTICAL NEIGHBORHOOD OPERATOR

Both linear and nonlinear neighborhood processors will be of interest to SDI for image manipulation if the operations can be fast enough to meet the realistic needs and if the computer satisfies other needs such as small cost, size, weight, and power consumption. Uses of neighborhood operations are either

- linear (sometimes called convolution, usually with Finite Impulse Response or FIR filters) or
- nonlinear (including shrink/expand and median operations).

We will concentrate on linear operations which can range from simple "derivative" operations such as the Sobel kernels

$$\begin{array}{ccc} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{array} \quad \text{and} \quad \begin{array}{ccc} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{array}$$

to convolution with pattern recognition templates.

This is a powerful technique which has inspired the production of several specialized ELU's (electronic processing units). Of these, by far the fastest are pipelined machines which flow the neighborhood ( $n \times n$ ,  $n$  odd) data into the ELU in a sequential manner so that calculating the next pixel requires dropping  $n$  data and adding  $n$  data. The fastest of these, the cytocomputer requires only 100 nsec per new pixel for  $n=3$ . Thus for TV resolution ( $0.25 \times 10^6$  pixels), we obtain a frame time of

$$\begin{aligned} t_F &= \frac{0.25 \times 10^6 \text{ pixels}}{\text{frame}} \cdot \frac{10^{-7} \text{ sec}}{\text{pixel}} \\ &= 0.025 \text{ sec/frame.} \end{aligned}$$

That is, the biggest and best is roughly "real time", but only for small images ( $500 \times 500$ ) and the smallest meaningful window ( $3 \times 3$ ). Calculation time scales

as  $N^2n^2$  for an  $N \times N$  image and an  $n \times n$  window. Furthermore we may need to cascade neighborhood operations  $M$  times, where  $M = 1$  to, say, 20. Thus the total image processing time is

$$t_I \approx MN^2n^2t_0,$$

where  $t_0$  is the basic operation time. Our dual objectives are to

- reduce the multiplier  $N^2n^2$  by increasing parallelism and
- reduce  $t_0$  by going to optics.

It is likely that, in the process, we will also improve on electronics in size, weight, cost, and power consumption. It is the speed increase, however, which should have the greatest impact on SDI, by allowing bigger images (larger  $N^2$ ) to be processed by more sophisticated algorithms (larger  $n$  and  $M$ ).

As currently designed our system will sacrifice a little in numerical accuracy relative to its electronic counterpart by operating analog rather than 16 bit fixed point. Renormalization after each of the  $M$  cycles will largely mitigate this problem.

So far, this introduction has emphasized our approach's advantages relative to electronics. We conclude the introduction by noting its advantage relative to most optical "computers". That advantage is that it can be built now with available components to perform tasks beyond current and foreseeable electronic capabilities. No further invention is required. No special components need to be developed.



REFERENCES:

1. "Interactive Image Processor Speeds Pattern Recognition" - K. Preston and P. E. Norgren, Electronics 45, 89 (1972).
2. "Language & Architecture for Parallel Image Processing", S. R. Sternberg in Pattern Recognition in Practice, E. S. Gelsema and L. N. Kanal, Eds., North Holland (Amsterdam), 1980.
3. "A Class of Cellular Architectures To Support Physical Design Automation", R. A. Rutenbar, T. N. Mudge, and D. E. Atkins, IEEE Trans. CAD of IC's and Systems CAD3, 264 (1984).
4. "Multiprocessor Architectures for Machine Vision and Image Analysis", R. M. Loughheed and D. L. McCubbery, Proc. Intl. Conf. on Parallel Processing, 493 (Aug., 1985).

## TECHNICAL DISCUSSION

### Introduction

Two-dimensional signal processing is a fundamental part of vision systems and is recognized as a speed bottleneck in contemporary pattern recognition where matched filters and a variety of 2-D transforms are needed. The following technical discussion is therefore focused on a few 2-D filtering systems and gives comparisons in terms of analog electronic versions using the same techniques. Finally a 2-D convolution filter using an acoustooptic cell is presented that closely approaches the ultimate data flow efficiency of one output data point per input image pixel.

#### A. LED-Mask Neighborhood Operator

Figure 1 shows nine LED's with a film filter mask, a collection lens, and a detector which multiplies the LED output (image pixels) by the film transmittance (filter weight) and collects the products on a single detector.

The advantage of this device is in providing instantaneous multiplication and addition. The addition feature is a "multiplexing" aspect of optics. No multiplexing is available at the input however, as one LED is required per input pixel. Further the LED's must be driven by individual linearized current sources that store the pixel's levels. This means that a data shifting mechanism is required on the input. This system is programmable by replacing the film filter function and has limited accuracy due to the characteristics of the LED's.

Figure 2 shows an analog electronic version of the LED-Mask device using individual resistors as the filter function, individual voltage sources as the input image pixel levels, and an output current signal which is a sum of products. This system is also essentially instantaneous in multiplication and summing. The addition occurs due to the multiplexing ability of current in wires which are tied together. No multiplexing occurs at the input where individual voltage sources are required that store the pixel values. Much higher accuracy is

2-D FILTER

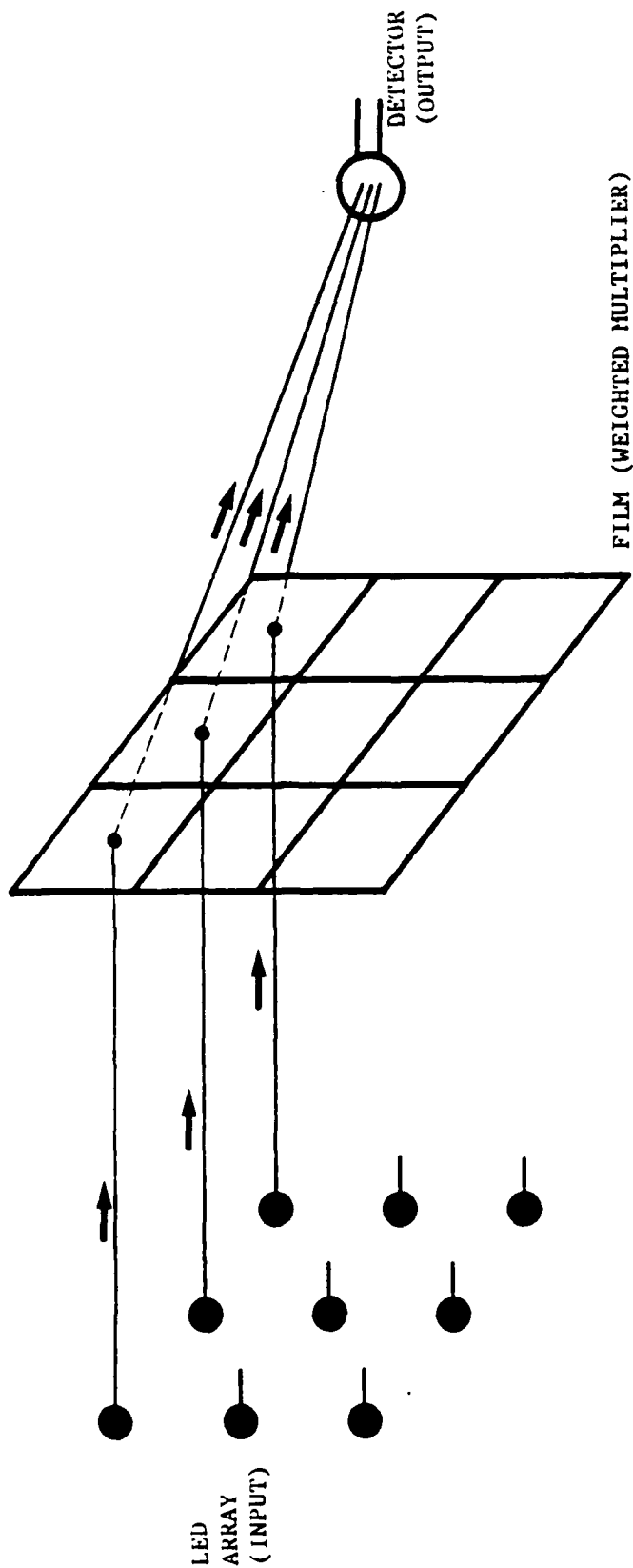


Figure 1.

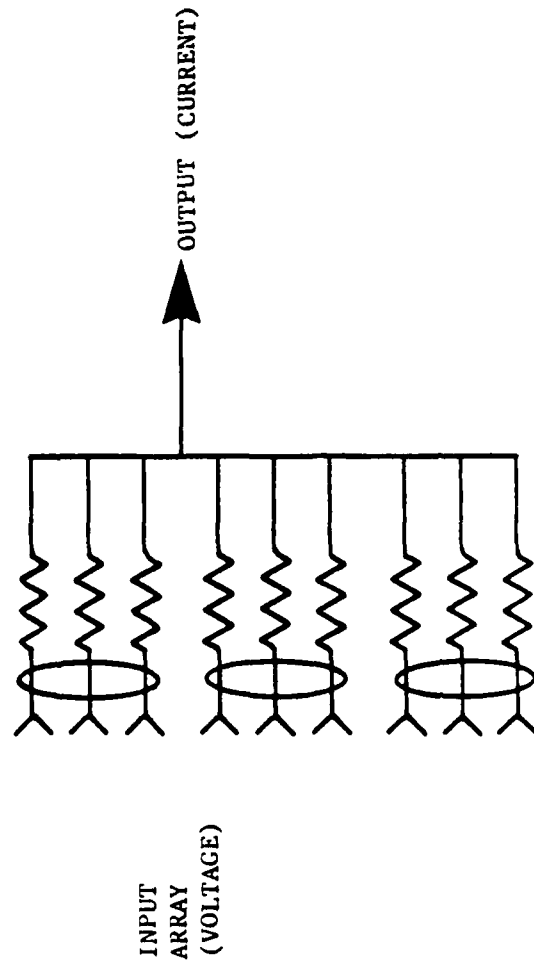


Figure 2.

possible, however, due to the elimination of the nonlinear LED's and the linear but not necessarily predictable optical efficiency of the parallel optical paths. The resistor multiplier is not programmable unless digital-to-analog converters are used, which do not add significant cost or complexity at this time.

In this previous case an analog electronic implementation had several advantages over the analog optical system, except possibly in the question of programmability. It is evident that the multiplexing aspect of optics was not limited only to optics in this case, as the electrical version was equally effective. We also observe that the lack of multiplexing at the LED input side of the optical device is inefficient, as it also is the electronic implementation.

A digital implementation of the neighborhood operator utilizes a digital multiplier and adder. The use of several multipliers to speed up the operation leaves the digital addition function as the bottleneck. Analog addition whether optical or electronic is very fast and may be preferable in this case, if analog electronic accuracy is not a problem in the application. One implementation of this mixed analog-digital idea would use digital-to-analog converters as multipliers, and collect the products by summing the output currents.

Analysis of the LED driven optical neighborhood operator has revealed several issues. First that the fast addition aspect of analog optics has its equivalent in analog electronic addition. Second that linearizing LED's is not as accurate and may be as complicated as using D/A converters in the analog electronic version, which makes it more easily programmed than using a replaceable filter function made of film. Third that the present digital electronic competition has accurate digital multipliers with ever increasing speeds and decreasing costs, and a speed bottleneck at the addition function.

#### B. Fourier Transform By Optical Matrix-Vector Multiplication

A discrete fourier transform maps a vector representing a discrete set of time samples, for example, into a vector whose components are the frequency bins.

This is a complex matrix-vector multiplication. This was demonstrated as shown in Figure 3 by Goodman, Diaz and Woody in 1978 using a series of LED's representing the input sample vector, a film mask and a set of detectors as the output frequency vector. The result is a very fast fourier transform involving  $N^2$  multiplications and  $N^2$  additions, where  $N$  is the number of LED's and detectors.

Analysis of this interesting device shows that the film mask is complicated, but easily replicated and that good use is made of the multiplexing aspect of optics both in the input and output of the device. This is possible here because a matrix-vector multiplication has  $N^2$  multiplication and  $N^2$  additions with only  $N$  inputs and  $N$  outputs. Other film masks can also be used providing different linear transformations.

This optical device also suffers the same accuracy problems as the LED neighborhood operator such as linearization of the LED's and inbalance among the many optical paths.

To assist in analyzing the optical matrix-vector multiplier we again present an analog electronic version as shown in Figure 4. The side view shows one column of the resistors which make up the resistor array which replaces the film mask. The input vector (voltages) is impressed on the rows of resistors and the output currents are collected along the columns. Addition is again accomplished by tying together wires to collect the output currents. Multiplexing occurs both at the input and output by tying together the rows and columns, respectively, on opposite sides of the resistor array.

The analog electronic version of the matrix-vector multiplier utilizes multiplexing at the input and output, demonstrates fast multiplication and addition, has good accuracy and can probably be built as an integrated circuit. The electronic version is not programmable as is the optical version, though the mask replacement may be as complicated as replacing the entire integrated electronic chip. The optical version may have the advantage that film can provide a medium for a matrix with over 1000 x 1000 points, while the integrated electronic version would require development. A matrix built

# FOURIER TRANSFORM

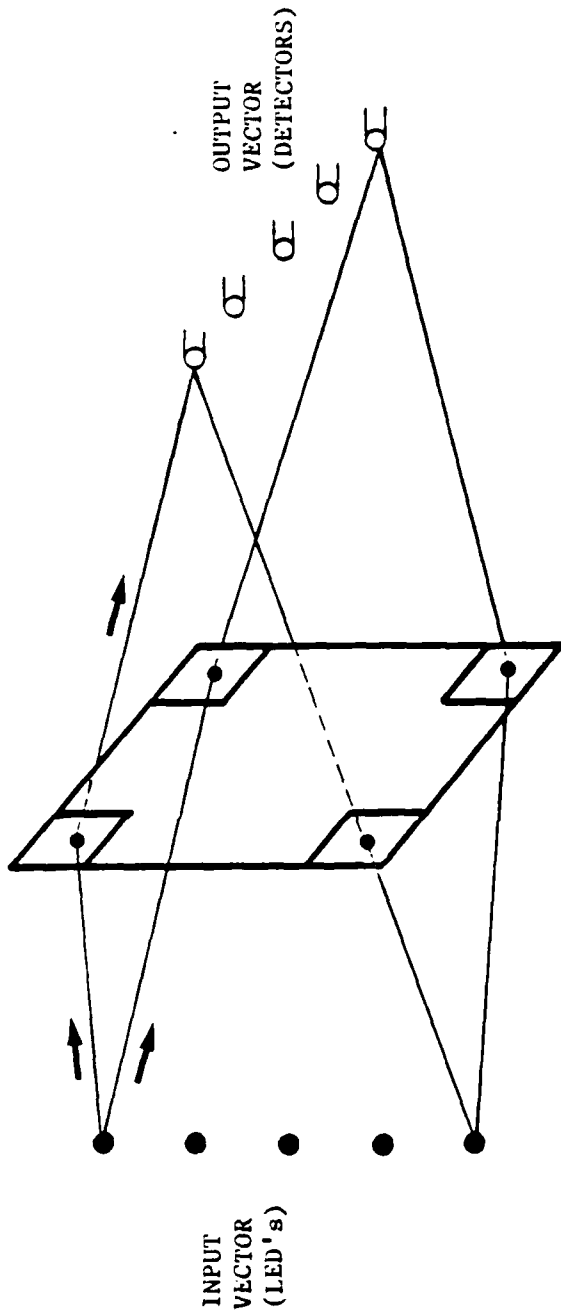


Figure 3.

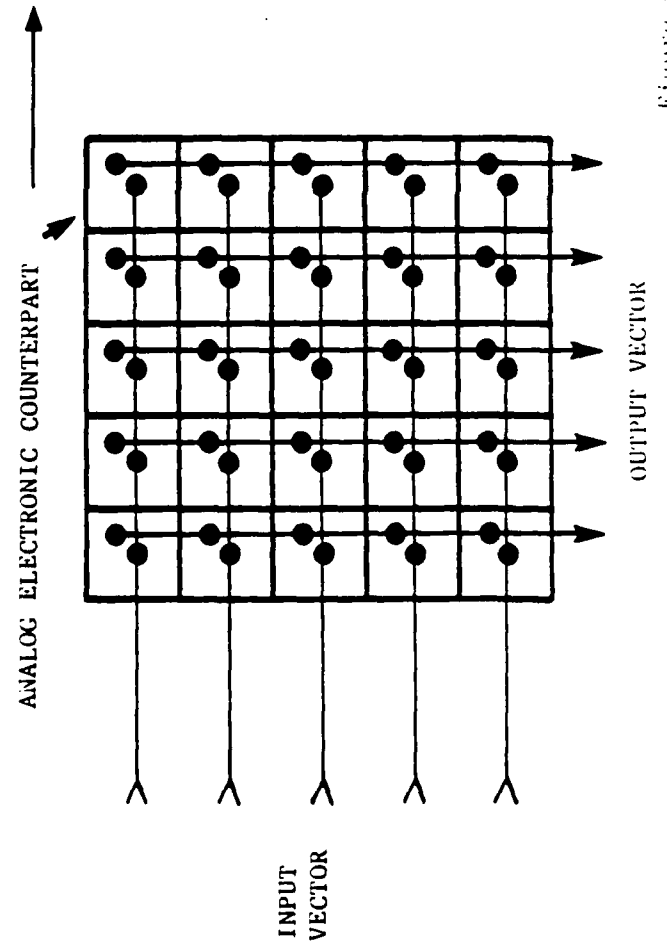
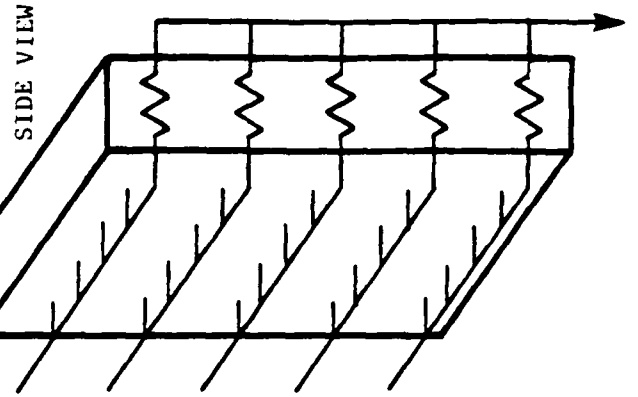


Figure 4.



A85-623

with discrete resistors would seem to be unreasonable with one million resistors, solder joints, etc.

Analyzing the optical matrix-vector multiplier by inventing an analog electronic version has brought out some of the advantages and disadvantages of using optics. The multiplexing and addition aspects of optics are notable but are not limited only to optics, in our cases. Especially desirable is the fact that replication of the film matrix for the fourier transform array would be straightforward, as it also would be for other useful linear transformations.

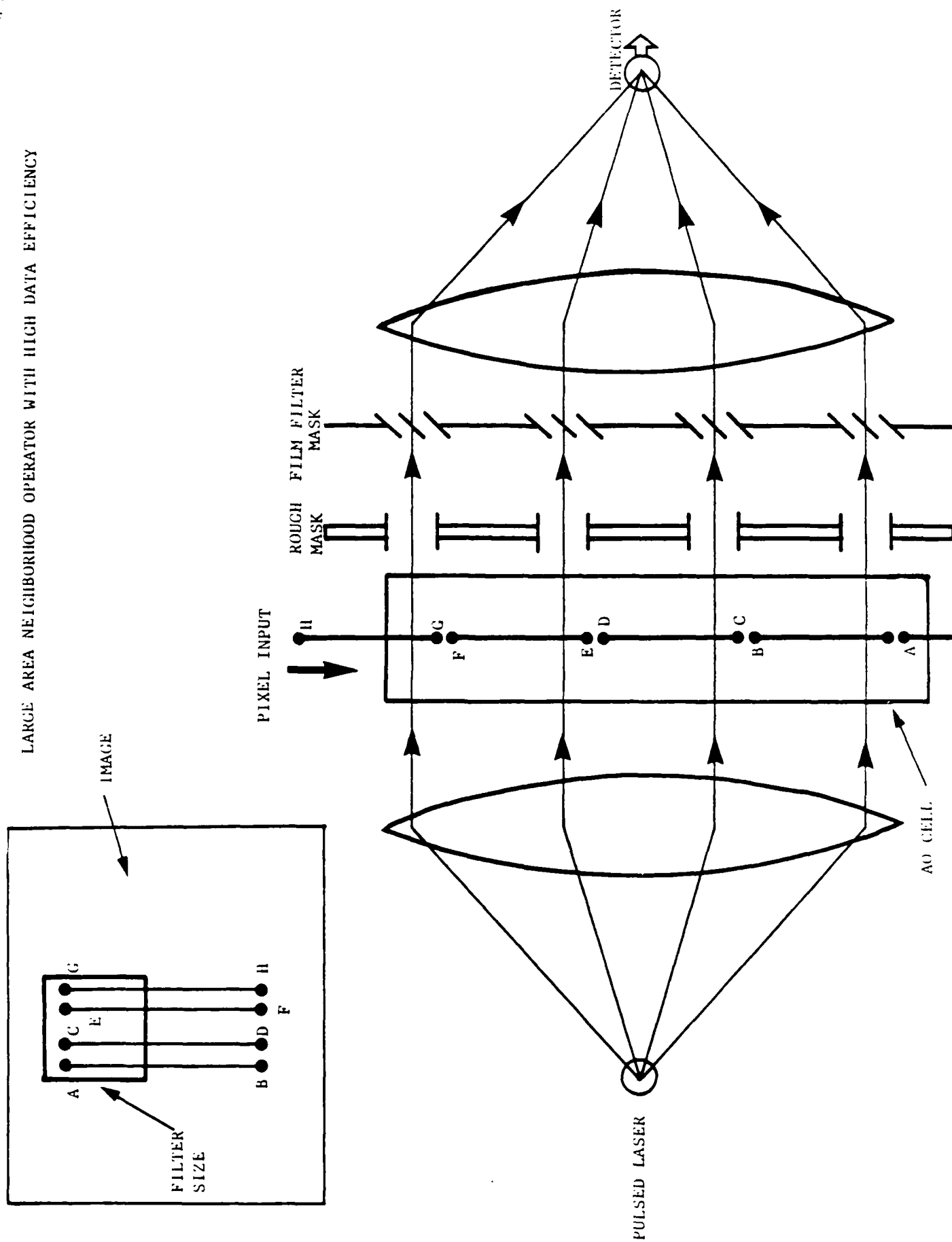
Implementing a discrete fourier transform by digital electronics involves many multiplications and additions. Specialized processors are now available which use several digital multipliers and are very efficient with data flow. However these would still be much slower than either the optical or electronic matrix systems that we have been analyzing.

C. 2-D Neighborhood Operator Using A Single Channel Acousto-Optic Cell Demonstrates Near Ultimate Data Flow Efficiency

This efficient image processing device can accomplish two dimensional convolution with a large filter function (for example  $9 \times 9$ ) using a single channel acoustooptic cell. Figure 5 shows the device which utilizes a film filter function mask and an AO cell to introduce the image information. The resulting optical signal is collected by a lens and focused on the output detector.

A digital electronic implementation involves multiplying 81 pixel levels ( $9 \times 9$  filter) by the filter weights, and adding them to give each output data point. In an inefficient system this involves recalling the image data 81 times. In a more efficient pipelined implementation, columns of data would be provided in steps, while shifting the columns inside the processor. This technique requires 9 input pixels per output data point, instead of 81, since one dimensional data flow is being used. If 2-D data flow were possible, one input pixel would yield, on the average, one output data point, which is the ultimate.

# LARGE AREA NEIGHBORHOOD OPERATOR WITH HIGH DATA EFFICIENCY





The AO cell 2-D convolver can approach within 10% of the ultimate data flow efficiency of one output data point per input pixel. This is accomplished by a pseudo-two-dimensional data flow in a one dimensional acoustooptic cell. The image enters the AO cell as a series of columns which are each 90 pixels long rather than the 9 pixels required by the desired  $9 \times 9$  convolution filter. Nine such extra long columns which sit side by side in the image, fit in tandem in the AO cell. Of the 810 pixels represented in the AO cell only 81 pixels will be used at any one moment. A mask adjoining the AO cell simultaneously selects the desired 9 pixels from each of the nine long columns and multiplies them by the film mask filter function. A pulsed laser source illuminates the AO cell, is multiplied by the filter function and is collected on the detector, giving a single neighborhood calculation. Each step of the column data through the AO cell gives a new output data point corresponding to the filter mask stepping down the long columns of the image. Ninety steps in the AO cell give 81 output points. After the ninety steps in the AO cell, the next column to the right in the image is inserted, step-by-step, into the AO cell. With the film mask adjoining the AO cell unmoved, all the columns in the AO cell are now facing a new set of filter weights, which corresponds to the filter function moving one pixel to the right on the original image. The result is efficient data flow in two dimensions, first by inserting one whole new column at a time, and second by making the columns much longer than required by the filter height. This implementation of a large two dimensional convolution filter requires each data point from the image only 1.1 times, on the average, which is within 10% of the ultimate data flow efficiency.

#### Analysis

The single channel AO cell neighborhood operator utilizes the large time-bandwidth product of the AO cell for data storage and shifting. Only a fraction of the AO cell is ever used to modulate light, which is inefficient. The data storage and shifting yields the near ultimate data flow efficiency, as is recognized by proposing a similar digital electronic convolver. In the digital

version extra long data delay lines are used feeding 81 multipliers and adders, requiring the image pixels from the main memory only 1.1 times per output data point. The digital system is complex with 81 digital multipliers and adders. The fact that addition is required at the end makes the digital system more complicated, and the optical system simpler, requiring only a lens and a detector. The fact that the data required for multiplication is in the same order as it is found in the image makes little difference to the digital system, but vastly simplifies the AO cell multiplier (optical modulator).

Accuracy decreases as the number of pixels stored in a given AO cell increases, and this trades off against data flow efficiency and filter size. This AO cell convolver is very wasteful of the AO cell accuracy, as mentioned before, as only about 10% of the cell is being used for modulating light. However the convenience of data storage, data shifting, and light modulation is quite notable, and well known.

Summarized below are the important characteristics of our AO cell convolver.

1. Near ultimate data flow efficiency (9 output data points per 10 input pixels).
2. Large filters handled (9 x 9).
3. Replaceable filter function (film).
4. Minimum number of sources and detectors.
5. Limited accuracy (20-30 dB estimated).
6. Requires unusual data sequence from image.
7. Practical now with off-the-shelf parts.
8. Ten times faster than desired video frame rate.
9. Extraordinary optical parallelism in an extremely simple device.

**RESEARCH ON OPTICAL COMPUTING ALGORITHMS AND ARCHITECTURES**

William T. Rhodes, Project Director and Principal Investigator  
Thomas K. Gaylord, Principal Investigator

GEORGIA INSTITUTE OF TECHNOLOGY  
SCHOOL OF ELECTRICAL ENGINEERING

Georgia Institute of Technology  
School of Electrical Engineering  
Atlanta, Georgia 30332

William T. Rhodes, Project Director and Principal Investigator  
Thomas K. Gaylord, Principal Investigator

**Research on Optical Computing Algorithms and Architectures**

*Abstract*

*Research on new architectural and algorithmic approaches to optical computing has been conducted in the areas of 1) optical degrees of freedom and devices for controlling them, 2) ultra-short optical pulses and nonlinear optics, 3) number representations, 4) content-addressable-memory processors, and 5) integrated optical Givens rotation devices. Results and recommendations are presented.*

## **Technical Summary**

### **1. Objectives**

The principal objective of this research program is the conceptual development of new architectural and algorithmic approaches to ultra-highspeed computing using optical and opto-electronic techniques.

### **2. Description of Work Performed and Results**

#### **2.1 Optical Degrees of Freedom and Devices for Controlling Them**

A review was conducted of the properties of light that can be controlled in an optical computer (e.g., polarization, propagation direction, wavelength, amplitude, phase, intensity), means for controlling them, and the advantages and disadvantages of different methods, including speed of operation and energy consumption. This was done to provide the basis for a study of optical computer architectures unprejudiced by notions of what basic light control operations should be employed and to provide to as great an extent as possible for flexibility in conceptual architectural design.

#### **2.2 Ultra-Short Optical Pulses and Nonlinear Optics [1]**

A preliminary study has been conducted of ways in which nonlinear optical phenomena can be used with ultrashort optical pulses to enhance the capabilities of optical computers. Ultrashort pulses are of interest because of their potential for exploiting the full available bandwidth of the optical source. It has been determined that nonlinear optical interactions can be used in various ways to allow for the cascade of highspeed content-addressable-memory-based optical computing systems and to compensate for loss and aberrations. Nonlinear optical interactions that exhibit both high speed and high efficiency (both of which are necessary for optical computer systems) are achievable only with optical pulses of high peak power. Mode-locked lasers and optical pulse compression systems produce such pulses every nanosecond or so. Unfortunately, schemes proposed thus far for exploiting the full temporal frequency bandwidth of the light source [2] result in reduced peak optical power and, hence, reduced efficiency in the nonlinear interactions. Attention is now being given to methods for avoiding this problem.

#### **2.3 Number Representation [3]**

Preliminary results have been obtained in the investigation of number representations for optical computing systems. Binary coding, multilevel coding, and residue number systems have been analyzed in terms of the primitive operations of addition and multiplication. Examples of fixed-radix and residue number representations have been calculated with and without multilevel coding. A detailed comparison has been made for the case of 16-bit full precision addition and multiplication. This example has indicated a clear advantage of

using multilevel coding.

#### 2.4 Content-Addressable Memory Processors [4]

Preliminary results have been obtained showing the use of optical content-addressable memory processors in non-primitive operations such as discrete matched filtering (cross correlation). The design of an optical holographic truth-table look-up system that processes multilevel coded numbers has been developed.

#### 2.5 Integrated Optical Givens Rotation Device [5]

The Givens rotation operation plays a central role in matrix formulations of linear algebraic signal processing. A design concept for an integrated optical device that implements this operation has been developed. The device uses electronically-controlled thick grating diffraction to control optical wave amplitudes in accord with the desired rotation operation. It has been shown that existing electro-optic phase shifting and grating diffraction devices can be combined to produce an extremely fast Givens rotation device. Operations that can be performed by such a device include matrix triangularization, matrix inversion, solution of least squares problems, singular value decomposition, and the calculation of eigenvalues and eigenvectors.

### 3. Conclusions and Recommendations

It is premature at this stage to draw many conclusions. However, in terms of program direction we think that the content-addressable-memory work is particularly important, for it shows promise for optical computer architectures capable of exploiting both the spatial and the temporal potential of optics. Particular attention should be given to multiple-input-multiple-output systems because of their significance in parallel processing generally.

The Givens rotation device is important of itself because of its possible applications. However, it is also significant because of the way it exploits natural physical phenomena for performing operations not easily performed on a binary-logic-based electronic computer. The basic approach discussed in an attachment needs to be studied further in terms of accuracy and speed achievable, and related architectures and algorithms should be investigated.

Nonlinear optics used in conjunction with ultrashort optical pulses can in principle solve many of the problems associated with wideband operation of cascable logic-based optical computer subsystems. There is, however, a conflict between the need for high peak-power optical pulses (for high-efficiency nonlinear interactions) and the temporal modulation of the light waves necessary for exploiting the full optical bandwidth. This conflict must be studied further and somehow resolved. Further, the energy and efficiency characteristics of nonlinear optical devices under development should be considered in connection with specific (e.g., strawmen) systems.

## References

1. W. T. Rhodes and J. A. Buck, "Optical Computing and Nonlinear Optics," to be presented at and appear in the proceedings of the SPIE conference 625 on Optical Computing, Los Angeles, January 23-24 1986. (In preparation.)
2. H. J. Caulfield and Tomas Hirshfeld, "Optical Communication at the Source Bandwidth Limit," **Applied Optics**, vol. 16, pp. 1184-1186, May 1977.
3. T. K. Gaylord and M. M. Mirsalehi, "Truth-Table Look-Up Processing: Number Representation, Multi-Level Coding, and Logical Minimization," **Optical Engineering**, vol. 25, pp. xxx-xxx, January 1986 (accepted). (Copy attached.)
4. M. M. Mirsalehi and T. K. Gaylord, "Multi-Level Coded Residue-Based Content-Addressable-Memory Optical Computing," submitted to **Applied Optics**. (Copy attached.)
5. M. M. Mirsalehi, T. K. Gaylord, and E. I. Verriest, "Integrated Optical Givens Rotation Device," submitted to **Applied Optics**. (Copy attached.)

Author: PLEASE read the proofs against your typewritten manuscript. You are responsible for verifying the accuracy of spelling, math, numerical data, the figures and legends, and references.

Issue date: January 1986  
Return proofs, ms., figures, & forms by  
Oct 23, 1985

Galley 1

OT-103/Gaylord

Opt. Eng. January 1986 mg

Disk 1 103-1

## Truth-table look-up processing: number representation, multilevel coding, and logical minimization

T. K. Gaylord, MEMBER SPIE  
M. M. Mirsalehi, MEMBER SPIE  
Georgia Institute of Technology  
School of Electrical Engineering  
Atlanta, Georgia 30332

**Abstract.** The need for ultra-high-speed computing for a variety of modern processing problems has generated new interest in using truth-table look-up techniques. Further, due to the frequently parallel nature of these processing problems, optical systems appear to be promising for these applications. The basic principles of truth-table look-up processing are reviewed in this paper. The issues of number representation, multilevel coding, and logical minimization are discussed. Example fixed-radix and residue number representations are given with and without multilevel coding. Logical reduction techniques are discussed with examples. A comparison of the number of truth-table entries needed for 16-bit full-precision addition and multiplication is given, illustrating the advantage of the multilevel coded residue number representation.

**Subject terms:** digital optical computing; truth-table look-up processing; optical data processing; number representation.

*Optical Engineering* 25(1), 000-000 (January 1986).



### Technical Discussion

Copies of references 3, 4, and 5 are attached.

Galley 2

OT-103/Gaylord

OE Jan 1986

Disk 1 103-2R

author: Has this paper been  
presented at any conference  
or published in any other  
publication? Editor

Invited Paper OT-103 received July 24, 1985; revised manuscript received Aug.  
28, 1985; accepted for publication Aug. 30, 1985; received by Managing Editor  
Sept. 23, 1985.  
© 1986 Society of Photo-Optical Instrumentation Engineers.

## CONTENTS

1. Introduction
  - 1.1. The need for ultra-high-speed computing
  - 1.2. Truth-table look-up processing as a possible solution
  - 1.3. Issues associated with truth-table look-up processing
2. Truth-table look-up processing architectures
  - 2.1. Location-addressable memories
  - 2.2. Content-addressable memories
  - 2.3. Hardware logic gates
3. Number representation
  - 3.1. Fixed-radix system
  - 3.2. Residue number system
4. Multilevel coding
  - 4.1. Encoding
  - 4.2. Example systems
5. Logical minimization
  - 5.1. Forms of logical reduction results
  - 5.2. Finding prime implicants
  - 5.3. Constructing table of choice
  - 5.4. Obtaining minimal sum
6. Optical implementation
7. Discussion and summary
8. Acknowledgments
9. References

## 1. INTRODUCTION

### 1.1. The need for ultra-high-speed computing

The number of areas in need of computing power well beyond that currently available is large and increasing. High throughput computing systems (or, equivalently, ultra-high-speed systems) are needed in areas such as adaptive antenna beam forming, artificial intelligence, remote sensing, ultra-high-resolution image processing, control of communication networks, aeronautical design, seismic data interpretation, meteorology, air-traffic control, synthetic-aperture radar imaging, missile guidance, defense early-warning systems, and molecular, nuclear, and plasma physics simulations.<sup>1-4</sup> For example, real-time computation of images from synthetic-aperture radar data would require the equivalent of several trillion multiplications per second.<sup>5,6</sup>

### 1.2. Truth-table look-up processing as a possible solution

Many functions, transformations, and operations may be represented by a binary truth table in which the outputs are given for all possible input combinations. Direct implementation of processors from a truth-table representation has not been common in the history of data and signal processing. This is largely due to the numerous efficient algorithms that can be programmed on general-purpose Von Neumann type computers. However, the types of problems listed above are largely beyond the capabilities of present-day computing systems. These problem areas have emphasized the growing need for parallel application of the same algorithm to large arrays of data. This, in turn, has generated renewed interest in the direct implementation of truth-table-based processors.

Many of these processing problems are highly complex and computationally intensive. However, the solutions can frequently be expressed in terms of matrix-based algorithms.<sup>7</sup> In these, a single operation is repeated many times over many elements. This highly regular nature lends itself naturally to parallel processing and to truth-table look-up techniques. The pronounced structure of the algorithms has not been efficiently utilized in past data processing systems.

There are three general architectures for truth-table implementation. These involve using (1) location-addressable memory, (2) content-addressable memory, and (3) hardware logic gates. These basic architectures are discussed in Sec. 2. Gate arrays and programmable array logic (PAL) devices that are in widespread use today are electronic implementations of truth tables. In another example, off-line a priori calculations are used to prestore in memory the controllers for given speed ranges for a fighter aircraft. This is necessary since the required calculations cannot be performed in real time and thus are obtained by look-up. Papachristou<sup>8</sup> has presented an encoding scheme for a direct truth-table implementation of discrete and residue-based functions (see Sec. 3.2) that employs PAL devices. Truth-table look-up has been used for changing the function in optical cellular logic to implement two-dimensional logical neighborhood functions for applications such as digital image processing.<sup>9</sup> Look-up methods have been used to find the correct mappings required to implement a residue matrix-vector multiplier.<sup>10</sup> Discrete matched filtering and other functions can be implemented by truth-table look-up techniques.<sup>11</sup> Ishihara<sup>12</sup> has described the use of truth-table look-up in the design of optical processing systems by the joint university-governmental-industrial Optical Computer Group of the Japanese Society of Applied Physics. Potentially, many complex problems can be treated with look-up methods.

mad /

i /

### 1.3. Issues associated with truth-table look-up processing

The viability of truth-table look-up processing for a particular application depends on a number of critical issues. These include (1) architectural implementation, (2) number representation, (3) number encoding, and (4) logical reduction and/or minimization. In an electronic or optical hardware logic gate implementation, the resulting number of gates and the number of interconnections are determined by the truth-table representation used. Perhaps more important, the needed routes of the interconnections are prescribed by the final form of the truth table used. In a bulk optical configuration using a content-addressable memory, the truth table used specifies the amount of storage needed (e.g., number of holographically recorded reference patterns).<sup>13</sup> For this case, however, the number and form of the interconnections specified are of no particular significance since the interconnections are made optically in three-dimensional space and their routing is automatically taken into account in the original design of the system. This is in dramatic contrast to very-large-scale integration (VLSI) in integrated circuits, in which the form of the interconnections is typically the limiting factor in the design of complex systems.

## 2. TRUTH-TABLE LOOK-UP PROCESSING ARCHITECTURE

### 2.1. Location-addressable memories

The most straightforward implementation of a truth table may be achieved by the storage of the entire truth table in a direct, or location-addressable, memory (LAM) such as an electronic read-only memory (ROM). These systems require a memory size (in bits) of

$$S = 2^p q, \quad (1)$$

where  $p$  is the number of input bits and  $q$  is the number of output bits. In these processors, the inputs determine the address of the answer.

### 2.2. Content-addressable memories

Less storage is generally required when a truth table is implemented using a content-addressable memory (CAM). Such memories may utilize electronic, magnetic, optical, or other technologies. The unity-result truth tables for each output bit are stored in the CAM. A unity-result or a null-result truth table may be constructed from those combinations of inputs that cause a particular output bit to be a "one" or a "zero," respectively. The unity-result truth table represents the canonical sum-of-products expression for the logical function corresponding to each output bit.

In a content-addressable memory, inputs are compared with the stored tables, and detected matches cause the appropriate output bits to be a "one" (if a unity-result truth table is stored) or a "zero" (if a null-result truth table is stored). The stored input words (or "reference patterns" in pattern recognition terminology) are the function minterms in the sum-of-products expression (unity-result truth table) or the function maxterms in the product-of-sums expression (null-result truth table). The number of function minterms for each output bit for addition and multiplication using the residue number system has been compiled.<sup>14</sup> This number is always less than or equal to the number of function maxterms due to the inherent structure associated with the operations of addition and multiplication. In the optical holographic implementation of content-addressable memory, the number of function minterms represents the number of holograms that need to be stored in the system.<sup>10</sup> Using thick holographic recording media, such as photorefractive lithium niobate, holograms may be multiplexed together in a common volume<sup>15</sup> with the number of possible stored holograms being on the order of a thousand.<sup>16</sup>

### 2.3. Hardware logic gates

A truth table may also be implemented through the direct use of Boolean logic gates. Each binary output variable, when represented as a sum of products (or product of sums) of binary input variables, may be implemented with three levels of logic in the form of a programmable array logic device. For a sum-of-products form, the sequence of logic gates is NOT, AND, OR. For a product-of-sums form, it is NOT, OR, AND. The number of function minterms represents the number of AND gates (in sum-of-products implementation) that must be formed to realize each output bit.

## 3. NUMBER REPRESENTATION

### 3.1. Fixed-radix system

In many ways, the manner in which numbers are represented places subtle and fundamental limitations on what types of calculations can be efficiently performed on them. One need only try to do calculations with Roman numerals to realize the impact and importance of the positional (Arabic) number system. Innovations such as the concept of zero as a position-holding digit, negative numbers, and the representation of fractions in a positional number system required centuries of evolutionary development.<sup>17</sup> It is only a sense of provincialism that causes the decimal system to be viewed as a uniquely appropriate number system. Historically, satisfactory progress in many areas of mathematics and engineering has been limited by number representation.<sup>18</sup> Number systems and number representation are again becoming the subject of increasing study for achieving faster and more efficient computation.

In a fixed-radix number system, any number  $N_b$  in a radix (base)  $b$  may be represented by  $a_n, \dots, a_{-m}$ , such that

$$N_b = \sum_{i=-m}^n a_i b^i. \quad (2)$$

where  $a_i$  is any of the  $b$  digits allowed. The integers  $n$  and  $m$  control the size and precision of the number. The base for the binary, octal, decimal, and hexadecimal number systems are 2, 8, 10, and 16, respectively. The ranges of digits are 0 and 1, 0 through 7, 0 through 9, and 0 through 9 and A through F, respectively. For example,  $1101_2 = 15_8 = 13_{10} = D_{16}$ .

A property of all fixed-radix number systems is the interdependence of digit results in numerical operations. In addition and multiplication this is manifested as a carry digit propagating from lower to higher significant digits. This requires that the most significant bit of a result cannot be known until calculation of all lesser significant bits has been completed. Thus, carry propagation represents a fundamental limitation for high-speed digital electronic data processing systems.

In truth-table look-up processing, all digits of the answer can be calculated simultaneously. However, carry propagation produces a very undesirable effect in these processors. Because the output digits depend on all lesser significant input digits, the truth tables can become enormous. As the number of input digits increases, the size of the resulting truth table increases exponentially. Clearly, a number system without interdigit dependence would be highly desirable to avoid these unmanageably large truth tables. The residue number system described below has no such interdigit dependence.

### 3.2. Residue number system

Unlike the commonly used decimal and binary number systems, the residue number system (RNS) is an unweighted system. The base of a residue system is chosen as  $n$  relatively prime (containing no common factors) numbers  $m_1, m_2, \dots, m_n$ , called moduli. Any integer  $X$  is then represented as an  $n$ -tuple  $(x_1, x_2, \dots, x_n)$ , where  $x_i = X_{\wedge m_i}$  (meaning  $X \bmod m_i$ ). This representation is unique if the range of  $X$  is less than or equal to  $M$ , where

$$M = \prod_{i=1}^n m_i. \quad (3)$$

and represents the dynamic range. Negative numbers can be included by an arbitrary partitioning of the range of the number system.

The important feature of RNSs is that the fixed-point arithmetic operations can be performed on each digit individually. That is, if  $X = (x_1, x_2, \dots, x_n)$  and  $Y = (y_1, y_2, \dots, y_n)$  are two numbers of the same system, then  $Z = X \circ Y = (z_1, z_2, \dots, z_n)$ , where  $z_i = (x_i \circ y_i)_{m_i}$  for  $i = 1, 2, \dots, n$ , and  $\circ$  represents the addition, subtraction, or multiplication operation. Division may be performed but it is difficult,<sup>19,20</sup> except for the remainder zero case.

As an illustrative example, consider a set of four moduli  $\{3, 4, 5, 7\}$ . In this system, the decimal numbers  $X = 23$  and  $Y = 14$  are represented as  $X = (2, 3, 3, 2)$  and  $Y = (2, 2, 4, 0)$ . The results of performing addition, subtraction, and multiplication on these numbers are  $X + Y = (1, 1, 2, 2)$ ,  $X - Y = (0, 1, 4, 2)$ , and  $X \times Y = (1, 2, 2, 0)$ , which are the residue representations of the correct answers, i.e., 37, 9, and 322, respectively.

In residue arithmetic, there are a number of basic operations that are difficult to perform. These are division, scaling, sign detection, overflow detection, and relative-magnitude determination. In spite of these difficulties, the fact that the calculations associated with different moduli are independent of each other makes RNSs suitable for parallel processing. An especially significant increase in the number of operations per second is achieved when the calculations are composed of residue addition and multiplication (e.g., in matrix-matrix multiplication or discrete Fourier transformation).

The cyclic nature of residue arithmetic makes it particularly suitable for optical implementations. Using the cyclic property of a phase of the polarization of light, a number of numerical optical residue processors have been developed.<sup>21-26</sup> Hughes Research Laboratories has constructed an electronic residue arithmetic digital image understanding system (RADIUS) that performs  $5 \times 5$  pixel generalized convolution operations on 8-bit pixels.<sup>27</sup> The moduli used are 31, 29, 23, and 19, the four largest primes less than  $2^5 = 32$ . Additions and multiplications are performed at high speed by truth-table look-up of the residues for each radix from a random-access memory (RAM). Binary-to-residue and residue-to-binary conversion are also accomplished by truth-table look-up.

Moduli selection is an important issue in the design of any system based on residue arithmetic. Many system parameters are affected by the moduli set, and conflicting requirements may make the selection difficult. For example, to reduce the execution time for all operations that involve a mixed-radix conversion, it is desirable to have as few moduli as possible; hence, large moduli are preferred. On the other hand, the hardware of most systems increases rapidly with the size of the moduli; therefore, using a large number of small moduli has the advantage of decreasing the complexity of the system.

There are other considerations that imply different selections,<sup>19</sup> such as increasing the storage efficiency, having unity multiplicative inverses, and having unity multipliers in the Chinese remainder theorem. A procedure for selecting the moduli that are optimum in the sense of requiring the minimum number of reference patterns is given in Ref. 14.

As mentioned previously, the lack of interdigit dependence makes RNSs potentially extremely useful in reducing the required number of reference patterns that need to be stored. This is a very powerful feature in content-addressable memory applications. For example, consider the addition of two 16-bit numbers. If the usual binary system is used to represent the numbers, a total of 36,507,222,016 reference patterns, each a 32-bit word, are needed to be stored in a CAM for truth-table look-up processing. However, using the moduli set {4,5,7,9,11,13}, the number decreases to only 694 patterns of 4-bit to 8-bit words. As shown in subsequent sections, further reduction in the number of reference patterns can be obtained by applying multilevel coding and logical minimization techniques.

#### 4. MULTILEVEL CODING

##### 4.1. Encoding

Multilevel coding has recently been used as a technique for further reducing the number of truth-table entries (reference patterns) that need to be stored.<sup>11</sup> Multilevel coding is an extension of binary coding in which more than two levels are used. For example, in three-level (ternary) coding, the integers zero to eight are represented as 00, 01, 02, 10, 11, 12, 20, 21, and 22, respectively. Minimization of multilevel coded reference patterns requires a type of logic different from the commonly used binary logic. The appropriate logic, known as multiple-valued logic, is an active area of research today.

Although significant progress has been made in the theoretical aspects of multiple-valued logic, there have been only a small number of electronic implementations of this logic. The first full-scale three-value electronic computer was completed in 1958 at Moscow State University in the Soviet Union.<sup>28</sup> Electronic ternary logic has been used in constructing arithmetic units.<sup>29</sup> Multiple-valued operation of integrated circuits has been investigated at the Naval Research Laboratory.<sup>30</sup> Currently, for example, the Intel 8087 floating-point processor uses four levels of current in ROMs. In optics, shadow-casting techniques have been used to implement multiple-valued logic.<sup>31</sup> The fact that there have not been more implementations is partly due to the difficulties in realizing multilevel devices and partly due to the significant progress that has been achieved in the area of binary logic systems. However, as has recently been shown<sup>11</sup> multilevel coding in some optical systems can be implemented as easily as binary coding.

##### 4.2. Example systems

Some examples of decimal, binary, residue, binary-coded residue, multilevel coded residue, and binary-coded multilevel-coded residue number representations are presented in Table I.



## 5. LOGICAL MINIMIZATION

### 5.1. Forms of logical reduction results

Procedures for the logical reduction of a truth table may produce results in a variety of forms. For logical functions expressed as a sum of products, these include (1) the near-minimal sum, (2) the minimal sum, and (3) the absolute minimal sum. A *near-minimal sum* is generally obtained by using a nonexhaustive reduction technique. In these methods, the sum-of-products logical expression is greatly reduced but not necessarily minimized. These techniques can be very fast computationally because they do not consider all reduction possibilities.<sup>32</sup> A *minimal sum* is a reduced sum-of-products expression that has the minimum possible number of terms in it. These forms are often called "sloppy minimal sums" in the literature.<sup>33</sup> An *absolute minimal sum* is a reduced sum-of-products expression that has both the minimum possible number of terms in it and the minimum number of factors (or variables) in each term. This form is also called the "real minimal sum" and sometimes (confusingly) the "minimal sum."

As an example, the logical minimization for the simple case of addition modulus 4 is shown in Fig. 1. The method illustrated in this figure represents only one particular approach to logical minimization. However, in general, the steps in minimization are (1) define the initial truth table for the operation in question; (2) find all prime implicants; (3) construct the table of choice; and (4) obtain a minimal sum. Each of these steps will be discussed in subsequent sections. An alternative technique, the use of the Karnaugh map,<sup>34</sup> allows a minimal sum to be obtained directly without using the steps listed here. It is a graphical method that allows the minimal sum to be visualized directly, but it is impractical for functions of more than about five variables. The steps listed above, however, can be programmed on a computer and can handle any number of input variables.

### 5.2. Finding prime implicants

As shown in Fig. 1, the function is first specified and then coded. In this case, since the modulus is 4, binary coding is used directly. From these results, the truth tables for the most significant bit (MSB) and least significant bit (LSB) may be constructed directly. Then the logical expressions may be written. All of the prime implicants<sup>35</sup> of a logical function can be determined by the Quine-McCluskey method.<sup>35,36</sup> This method is summarized in numerous textbooks<sup>37</sup> and is illustrated in Fig. 1 for the MSB. The minterms are listed in subgroups starting with those that have a single "one" in them, then those with two "ones" in them, and so on, until all minterms are listed in the first group. Then, all pairs of minterms that differ by only one factor are checked. These pairs are listed in a second group with "don't care" dashes at the location of the differing factor for the pair. The process of combining terms that differ by only one factor is then continued until no further combining is possible. For the example in Fig. 1, no further combining is possible in the second group. All unchecked terms in the prime implicant table constitute the list of all prime implicants.

21  
21

32

PS

FE L A A 2

As the number of variables increases, the Quine-McCluskey method of determining the prime implicants becomes inefficient in terms of execution time and required memory. More efficient methods include the Tison algorithm<sup>33</sup> and the tree-structured approach of Morreale.<sup>34</sup> An even more efficient modified tree structure method has been developed recently by Guest.<sup>35</sup>

### 5.3. Constructing table of choice

In the Quine-McCluskey method, after the prime implicants have been determined, a table of choice is constructed. This consists of all the prime implicants (listed vertically in Fig. 1) and all the function minterms (listed horizontally in Fig. 1). Each prime implicant row is marked in the columns of the minterms covered by that prime implicant. Thus, it is observed how the entries in the initial truth table are covered by the prime implicants.

### 5.4. Obtaining minimal sum

A minimal sum is obtained by finding the minimal prime implicant covering of the table of choice. This is referred to in the literature as the covering problem, the set-covering problem, or the minimum-covering problem.<sup>33</sup> An absolute minimal sum may be found using the following steps. Note that this absolute minimal sum may not be unique; there may be other absolute minimal sums that can be obtained by changing the order in which the selections below are made.

*Step one:* Select essential rows. Some rows uniquely cover some of the columns. These rows must be selected in order to cover those columns. The prime implicants associated with these rows are called essential prime implicants. The essential rows and all columns that contain marks in these rows should be eliminated from the table of choice.

*Step two:* Eliminate dominated rows. One row dominates a second row if the first row has marks in all columns in which the second row has marks. If the dominating row has the same or fewer variables in its prime implicant, the prime implicant associated with the dominated row should be eliminated from the table of choice. If a minimal sum, rather than an absolute minimal sum, is satisfactory, then the number of variables in the prime implicants need not be compared.

*Step three:* Eliminate dominating columns. Similarly, one column dominates a second column if the first column has marks in all rows in which the second column has marks. The minterms associated with the dominating columns should be eliminated from the table of choice.

*Step four:* Repeat steps one through three until all columns are eliminated. The resulting sum of all essential row prime implicants represents the absolute minimal sum.

There are cases, however, in which the above procedure is unable to eliminate all columns. The remaining table, which contains at least two marks in each column, is called a cyclic table. In this case, the tabular method using a recursive branch-and-bound algorithm presented by Muroga<sup>33</sup> may be used to obtain the absolute minimal sum.

ease verify this wording.

Minimization techniques in multiple-valued logic are somewhat different from those used in binary logic. In binary logic, if two terms in a sum-of-products expression are the same in all bit positions except one, they can be combined into one term that has a "don't-care" bit at that location. For example, 100 and 101 can be combined as 10X, where X represents a "don't-care" bit. In multiple-valued logic, terms can be combined in several ways. For example, in ternary logic, the terms 120, 121, and 122 can be reduced to 12X, where X (referred to as a "complete-don't-care" digit) represents a digit with possible values of 0, 1, and 2. If one of the above terms is absent, the other two can still be combined. For example, the terms 120 and 121 can be reduced to  $12X_{01}$ , where  $X_{01}$  (referred to as a "partial-don't-care" digit) represents a digit with possible values of 0 and 1, but not 2.

As the number of entries in a truth table increases, the minimization procedure becomes too complex to be handled by hand. Associated with the present work, a computer program has been developed to reduce the reference patterns for an arbitrary level coding and to obtain the minimum number of required patterns. The Quine-McCluskey technique was extended to handle the multiple-valued logic case. In the first part of the program, a complete list of the prime implicants is obtained. Using this set, a table of choice is constructed. Then, a minimal sum set is obtained by applying the reduction rules to the table. The results for residue addition and multiplication for moduli 2 through 32 are given in Ref. 11. These results show that the number of reference patterns can be decreased significantly if the appropriate level of coding is used. If the modulus can be expressed as  $M = p^n$ , where  $p$  is a prime number and  $n$  is a positive integer greater than one,  $p$ -level coding is the best choice. For example, binary coding is appropriate for moduli such as 4, 8, 16, and 32, while ternary coding is beneficial for moduli such as 9 and 27. This is due to the highly regular structures of the truth tables that are produced in these cases. For a modulus that is not expressible in the above form, the proper coding level can be found among its prime factors. The prime factor that produces the largest contribution to the modulus is usually the best choice. For example, binary coding is appropriate for modulus 12 ( $= 2^2 \times 3$ ), while modulus 6 ( $= 2 \times 3$ ) benefits from ternary coding.

The optimum sets of moduli for minimizing the number of reference patterns for performing 16-bit full-precision addition and multiplication are given in Table II. Results for both binary-coded residue numbers and multilevel coded residue numbers are given before and after logical minimization.

## 6. OPTICAL IMPLEMENTATION

An optical implementation of a truth-table look-up data processing system using a holographic content-addressable memory is described in Ref. 11. The optical system presented is capable of processing multilevel-coded numbers. The operations of addition, multiplication, and discrete matched filtering (cross-correlation) are evaluated in terms of the number of required reference patterns for various word lengths.

## 7. DISCUSSION AND SUMMARY

Truth-table look-up processing concepts, implementations, and applications have been reviewed. Due to current and future needs for ultra-high-speed computing, there has been an increasing interest in using truth-table look-up techniques. Further, due to the parallel nature of numerous modern processing problems, optical systems are serious candidates for these applications.

The issues of number representation, multilevel coding, and logical minimization have been discussed. The number of entries (reference patterns) in the reduced truth table is of central importance in determining the viability of look-up techniques for a particular application. In hardware logic gate implementations (electronic or optical), the number of gates and the number of interconnections are prescribed by the logically reduced form of the truth table used. In the location-addressable memory and content-addressable memory implementations, the reduced truth table specifies the amount of storage required. In these cases, the interconnections are of no particular significance, in marked contrast to the hardware logic gate case. In an optical content-addressable memory, the truth-table look-up processing can be performed in parallel.

For comparison, the number of truth-table entries for 16-bit full-precision addition and multiplication are given in Table III. Results are supplied for binary and residue representations. Values are given with and without logical minimization and with and without multilevel coding. The dramatic reduction using residue number systems is apparent. Further significant reductions are shown by using logical minimization and multilevel coding. Thus, number representation, multilevel coding, and logical minimization are all significant factors in truth-table look-up processing.

## 8. ACKNOWLEDGMENTS

This work was supported in part by a grant from the Strategic Defense Initiative Office, administered through the Office of Naval Research, and by a grant from the Joint Services Electronics Program.

Opt. Eng. now includes titles of journal and proceedings papers and book chapters in the references. If readily available, on a separate page please provide titles for each reference.

Galley 13

OT-103/Gaylord

Opt. Eng. January 1986 mg

Disk 1 103-13

re authors needed for Refs. 2-4? Are page numbers needed for Refs. 2 & 4?

## 9. REFERENCES

1. A. L. Robinson, *Science* 203(4376), 156 (1979).
2. *Computer* 16(6), (1983).
3. *Phys. Today*, special issue on Advances in Computers for Physics, 37(5), 61 (1984).
4. *Proc. IEEE*, special issue on Supercomputers—Their Impact on Science and Technology, 72(1), (1984).
5. W. M. Brown, *IEEE Trans. Aerospace Electron. Sys.* AES-3(2), 217 (1967).
6. W. M. Brown, G. G. Houser, and R. E. Jenkins, *IEEE Trans. Aerospace Electron. Sys.* AES-9(2), 166 (1973).
7. K. Bromley, "An interview with Keith Bromley on signal processing," *Optical Engineering Reports No. 14*, p. 1, SPIE (February 1985).
8. C. A. Papachristou, *IEEE Trans. Comput.* C-32(10), 961 (1983).
9. T. Yatagai, S. Inaba, H. Nakano, and M. Suzuki, "Automatic flatness tester for very large scale integrated circuit wafers," *Opt. Eng.* 23(4), 401 (1984).
10. S. F. Habiby and S. A. Collins, *Topical Meeting on Optical Computing, Technical Digest*, pp. TuD41-TuD44, OSA, Washington (1985).
11. M. M. Mirsalehi and T. K. Gaylord, submitted to *Appl. Opt.*
12. S. Ishihara, *Topical Meeting on Optical Computing, Technical Digest*, pp. TuE21, OSA, Washington (1985).
13. C. C. Guest and T. K. Gaylord, *Appl. Opt.* 19(7), 1201 (1980).
14. C. C. Guest, M. M. Mirsalehi, and T. K. Gaylord, *IEEE Trans. Comput.* C-33(10), 927 (1984).
15. J. E. Weaver and T. K. Gaylord, "Evaluation experiments on holographic storage of binary data in electro-optic crystals," *Opt. Eng.* 20(3), 404 (1981).
16. D. L. Staebler, W. J. Burke, W. Phillips, and J. J. Amodei, *Appl. Phys. Lett.* 26(4), 182 (1975).
17. J. R. Newman, *The World of Mathematics*, pp. 430-520, Simon & Schuster, New York (1956).
18. R. M. Kline, *Digital Computer Design*, p. 19, Prentice-Hall, Englewood Cliffs, NJ (1977).
19. N. S. Szabo and R. I. Tanaka, *Residue Arithmetic and Its Applications to Computer Technology*, McGraw-Hill, New York (1967).
20. E. Kinoshita, H. Kosako, and Y. Kojima, *IEEE Trans. Comput.* C-22(2), 134 (1973).
21. A. Huang, in *Proc. Int. Optical Computing Conference*, pp. 14-18, IEEE, New York (1975).
22. A. Huang, Y. Tsunoda, J. W. Goodman, and S. Ishihara, *Appl. Opt.* 18(2), 149 (1979).
23. A. Tai, I. Cindrich, J. R. Fienup, and C. C. Aleksoff, *Appl. Opt.* 18(16), 2812 (1979).
24. S. A. Collins, Jr., "Numerical optical data processor," in *Effective Utilization of Optics in Radar Systems*, B. W. Vatz, ed., *Proc. SPIE* 128, 313 (1977).
25. J. N. Polky, D. D. Miller, and R. L. Gutmann, "Optical residue arithmetic data processing," Boeing Aerospace Company, Report AFWAL/AADO-2, Wright Patterson AFB, Ohio (1982).
26. J. Jackson and D. Casasent, *Appl. Opt.* 22(18), 2817 (1983).
27. S. D. Fouse, G. R. Nudd, and A. D. Cummings, in *Proc. 6th Int. Conf. on Pattern Recognition*, pp. 262-269, IEEE, New York (1982).
28. D. C. Rine, ed., *Computer Science and Multiple-Valued Logic*, North-Holland, Amsterdam (1977).
29. P. Sebastian and Z. G. Vranesic, in *Proc. 1972 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, New York (1972).
30. G. Abraham, *Computer* 7(9), 42 (1974).
31. R. Arrathoon and S. Kozaitis, "Shadow-casting for multiple-valued associative logic," *Opt. Eng.* 25(1), (1986).
32. Z. Arevalo and J. G. Bredeson, *IEEE Trans. Comput.* C-27(11), 1028 (1978).
33. S. Muroga, *Logic Design and Switching Theory*, p. 163, John Wiley & Sons, New York (1979).
34. M. Karnaugh, *IEEE Trans. Commun. & Electron.* 72(5), 593 (1953).
35. W. V. Quine, *Am. Math. Monthly* 59(10), 521 (1952).
36. E. M. McCluskey, *Bell Sys. Tech. J.* 35(6), 1417 (1956).
37. T. L. Booth, *Digital Networks and Computer Systems*, John Wiley & Sons, New York (1971).
38. E. Morreale, *IEEE Trans. Electron. Comput.* EC-16(5), 611 (1967).
39. C. C. Guest, "Holographic optical digital parallel processing," Ph.D. thesis, Georgia Institute of Technology (1983).

103-13

AIEE

Fig. 1. The process of logical minimization for residue addition modulus 4 is illustrated, including the steps of defining the initial truth table, finding all prime implicants, constructing the table of choice, and finding the absolute minimal sum.

Opt. Eng. January 1986 mg

OT-103/Gaylord

Galley 14

Disk 1 103-14

TABLE I. Number Representation Examples

Number representation	Decimal	Binary	Residue	Binary-coded residue	Multilevel coded residue	Binary-coded multilevel coded residue
Digit weight	10 1	64 32 16 8 4 2 1				
Moduli	-	-	9 5 4	9 5 4	9 5 4	9 5 4
Coding level	-	-	-	-	3 5 2	3 5 2
Digit maximum value	99	1111111	8 4 3	1000 100 11	22 4 11	10 10 100 1 1
A	7	0000111	7 2 3	0111 010 11	21 2 11	10 01 010 1 1
B	14	0001110	6 4 2	0101 100 10	12 4 10	01 10 100 1 0
A + B	21	0010101	3 1 1	0011 001 01	10 1 01	01 00 001 0 1
A x B	98	1100010	8 3 2	1000 011 10	22 3 10	10 10 011 1 0

Paste up.  
Run back, delete  
level, and  
expand under line

FL

underline /

Script. line ///

4/

\* Addition of two 16-bit words produces a 16-bit sum with an output carry bit (no input carry bit). Multiplication of two 16-bit words produces a full 32-bit product (rather than the simpler fixed-point or floating-point representations).

Galley 16

OT-103/Gaylord

PE Can Be 103

Disk 1 103-16R

TABLE III. Comparison of Number of Required Reference Patterns to Perform 16-Bit Full-Precision Addition and Multiplication Using Various Encoding Schemes\*

FL /

	Addition	Multiplication
Binary (without logical reduction)	$3.65 \times 10^{10}$	$6.32 \times 10^{10}$
Binary (after logical minimization)	$3.28 \times 10^6$	$1.43 \times 10^7$
Binary-coded residue (without logical reduction)	694	3252
Binary-coded residue (after logical minimization)	327	1183
Multilevel-coded residue (without logical reduction)	568	2540
Multilevel-coded residue (after logical minimization)	300	1067

\*Two-, three-, and five-level coding have been used in the multilevel coding results.



# BINARY-CODED RESIDUE ADDITION MODULUS FOUR

## RESIDUE ADDITION MODULUS FOUR

$x$	0	0	0	0
$+y$	0	1	2	3
$z$	0	1	2	3
	0	1	2	3
	1	0	3	2
	2	3	0	1
	3	2	1	0



$$\begin{array}{r} a_2a_1 \\ +b_2b_1 \\ \hline c_2c_1 \end{array}$$

$a_2a_1$	00	01	10	11
$b_2b_1$	00	01	10	11
$c_2c_1$	00	01	10	11
	00	01	10	11
	01	10	11	00
	10	11	00	01
	11	00	01	10



$$a_2a_1b_2b_1$$

## TRUTH TABLE

$c_2 = 1$	$c_1 = 1$
0	0
0	0
0	0
0	1
1	0
1	0
1	1
1	1



Fig. 1 (top portion)

# LOGICAL EXPRESSIONS

$$\begin{aligned} &\overline{a_2}\overline{a_1}b_2\overline{b_1} + \overline{a_2}\overline{a_1}b_2b_1 + \overline{a_2}a_1\overline{b_2}b_1 + \\ &\overline{a_2}a_1b_2\overline{b_1} + \overline{a_2}a_1b_2b_1 + a_2\overline{a_1}\overline{b_2}b_1 + a_2\overline{a_1}b_2\overline{b_1} + \\ &a_2a_1\overline{b_2}b_1 + a_2a_1b_2\overline{b_1} = c_2 \\ \\ &\overline{a_2}\overline{a_1}\overline{b_2}b_1 + \overline{a_2}\overline{a_1}b_2b_1 + \overline{a_2}a_1\overline{b_2}b_1 + \\ &\overline{a_2}a_1b_2\overline{b_1} + \overline{a_2}a_1b_2b_1 + a_2\overline{a_1}\overline{b_2}b_1 + \\ &a_2a_1\overline{b_2}b_1 + a_2a_1b_2\overline{b_1} = c_1 \end{aligned}$$

## PRIME IMPLICANT TABLE $c_2 = 1$

### FIRST GROUP

$a_2a_1b_2b_1$			
0 0 1 0 ✓			
1 0 0 0 ✓			
<hr/>			
0 0 1 1 ✓			
0 1 0 1			
0 1 1 0 ✓			
1 0 0 1 ✓			
1 1 0 0 ✓			
<hr/>			
1 1 1 1			

### SECOND GROUP

0	0	1	-
0	-	1	0
1	0	0	-
1	-	0	0
<hr/>			

## PRIME IMPLICANTS $c_2 = 1$

$a_2a_1b_2b_1$
0 1 0 1
1 1 1 1
0 0 1 -
0 - 1 0
1 0 0 -
1 - 0 0

Fig. 1 (upper middle portion)

[illegible]

Fig. 1 (lower middle portion)

# REDUCED LOGICAL EXPRESSIONS

$$\overline{a_2} \overline{a_1} b_2 + \overline{a_2} b_2 \overline{b_1} + a_2 \overline{b_2} \overline{b_1} +$$

$$a_2 \overline{a_1} \overline{b_2} + \overline{a_2} a_1 \overline{b_2} b_1 + a_2 a_1 b_2 b_1 = c_2$$

$$a_1 \overline{b_1} + \overline{a_1} b_1 = c_1$$



## REDUCED TRUTH TABLE

	$c_2 = 1$	$c_1 = 1$
$a_2 a_1 b_2 b_1$	0 0 1 X 0 X 1 0 1 X 0 0 1 0 0 X 0 1 0 1 1 1 1 1	X 1 X 0 X 0 X 1

Fig. 1 (bottom portion)

Truth-Table Look-Up Parallel Data Processing  
Using an Optical Content-Addressable Memory

M. M. Mirsalehi and T. K. Gaylord  
School of Electrical Engineering  
Georgia Institute of Technology  
Atlanta, Georgia 30332

Truth-table look-up data processing using a holographically implemented content-addressable memory is described. The implementations of the addition, multiplication, and discrete matched filtering (cross-correlation) are presented. It is shown that multi-level coding can be used to reduce the number of reference patterns in content-addressable memories. The number of reference patterns required to implement residue addition and multiplication operations are provided for moduli 2 through 32 and for 2-, 3-, and 5-level coding. An optical truth-table look-up processor based on multi-level coding scheme is presented.

1. Introduction

A. The Need for Optical Digital Computing

In spite of recent advances in electronic computers, there exists a number of problem areas<sup>1</sup> such as meteorology, aerodynamics, molecular dynamics, fusion energy, and finite-element analysis, that demand computing powers well beyond what is currently available. The major difficulty (known as the von Neumann bottleneck<sup>2</sup>) in increasing the computing power of existing electronic computers, is the sequential nature of the processing. To overcome this difficulty, parallel processing architectures are under development for the next generation of computers. Optical processors, in contrast to electronic systems, are inherently parallel; they can perform operations on the elements of a two-dimensional array, simultaneously. Optical digital systems combine the parallelism and speed of optics with the accuracy and flexibility of a digital system. As such, they are promising candidates for increasing computing power.

B. Associative and Content-Addressable Memories

In precise computer engineering terminology<sup>3</sup>, an associative memory and a content-addressable memory have different capabilities. An associative memory allows recall with incomplete or imperfect input information. It also takes into account input contextual information that may be present and provides a single output answer. A content-addressable memory, on the other hand, implements a type of associative recall. However, the recall is based on a complete parallel search of all reference patterns stored in the memory. Thus, the output can be a parallel array of answers, as opposed to a single output answer in the case of an associative memory. Thus, in general,

a content-addressable memory is considered to be a single-input/parallel-output system.

With optics, a content-addressable memory can be implemented in a straightforward manner by holographically recording many reference patterns using the large dynamic range associated with volume holographic recording media such as electro-optic crystals<sup>4</sup>. In addition, however, the basic single-input/parallel-output character of a content-addressable memory can be expanded. The lack of angular selectivity in the direction perpendicular to the recording plane of incidence of volume holograms<sup>5</sup> allows an entire array of parallel inputs to be processed simultaneously by the same set of holographic gratings. An optical volume holographic content-addressable memory can produce a separate set of parallel outputs for each of the inputs. Therefore, such a memory is capable of performing as parallel-input/parallel-output system. This, in turn, produces the possibility of very high throughput parallel computing.

### C. Truth-Table Look-Up Processing

Digital operations can be implemented by 1) converting the operation into a sequence of logic steps that can be realized by logic gates, or 2) reading the output from a memory in which the truth table corresponding to that operation has been previously stored. In spite of the superiority of the latter method in speed and flexibility, it has not been common in the history of data processing. This is mainly due to the large sizes of the memories that are usually needed. Recent technological advances and a growing need for parallel processing have generated renewed interest in the direct implementation of truth-table look-up processing.

There are two methods of implementing a truth table. One method

is to store each output bit in a memory location whose address is determined by the input word. This type of memory is called a location-addressable memory (LAM). In the second method, for each output bit, all the input bit combinations that produce a "one" in that output bit location are stored. During the processing step, the input bits are compared with all the stored reference patterns that correspond to each output bit. If a match is detected, that particular output bit is considered to be a "one", otherwise, it is a "zero". This type of memory is called a content-addressable memory (CAM). Content-addressable memories can benefit from logical reduction techniques, hence they usually require much less storage than LAM's.

As mentioned above, one of the advantages of the truth-table look-up technique is its flexibility. Any discrete function or operation whose truth table is known can be implemented by this technique. This includes addition, multiplication, division, exponentiation, series evaluation, and other operations. For example, truth-table look-up processing may be used to implement discrete matched filtering. Consider a one-dimensional discrete reference signal,  $g(n)$ , of four units in length, defined as:  $g(n) = n$  for  $1 \leq n \leq 4$ . Assume that it is desired to detect the signals similar to  $g(n)$ . First, all possible input signals are normalized so that the resulting discrete signals,  $f(n)$ , have energies that are the closest possible values to (but not greater than) the energy of the reference signal,  $g(n)$ . Second, the cross-correlations between the normalized signals,  $f(n)$ , and the reference signal are obtained (Fig. 1). If the two signals are the same, then the cross-correlation operation becomes an autocorrelation and a maximum peak value is produced. If the signals are different, the result will have a lower peak value of its cross-correlation. The height of the peak depends on the

similarity of the two signals. For the above example, the autocorrelation has a peak of 30. It might be of interest to detect not only the exact matched pattern, but also the patterns that have a high cross-correlation (for further inspection). In this case, in addition to  $g(n)$ , these patterns are also stored in the content-addressable memory. For example, if patterns of four units in length that produce a cross-correlation peak equal to or greater than 28 are desired, then 12 reference patterns need to be stored. The output of the system can be represented by a two-bit number. An exact match could be indicated by making the first bit equal to a "one", and a mismatch with a high cross-correlation peak (28 or 29 in the above example) could be indicated by making the second bit equal to a "one". In the case of a mismatch with low cross-correlation peak, both output bits would be "zero". Thus all possible input signals can be checked in a cross-correlation sense using only a few stored reference signals (12 in the above example). This is representative of truth-table look-up processing: a large number of calculations are done in advance in order to construct a relatively small truth-table that can then be used repeatedly to perform a calculation on all possible future inputs.

## 11. Truth-Table Representation

### A. Truth-Table Reduction

As mentioned in the previous section, the major problem that faces the truth-table look-up processing technique is the large number of reference patterns that usually need to be stored. This has prevented the implementation of electronic look-up processors except for simple cases. Although, holographic memory systems with large storage capacities exist, the amount of data storage required by

CAM's is still typically larger. For example, to implement the full-precision addition of two 16-bit numbers with a CAM based on the usual binary number system, a total of 36,507,189,248 reference patterns are required to be stored. This is dramatically beyond the number of holograms that can be recorded in state-of-the-art holography. Two methods of reducing the number of reference patterns that have been previously investigated are: 1) Using residue number system (RNS)<sup>4</sup>, and 2) applying logical minimization techniques<sup>6</sup>.

### B. Residue Number System

A residue number system is defined by choosing  $n$  relatively prime numbers,  $m_1, m_2, \dots, m_n$ , called moduli. Any integer  $X$ , can then be represented as an  $n$ -tuple,  $(x_1, x_2, \dots, x_n)$ , where  $x_i = |X|_{m_i}$  (read  $X$  modulo  $m_i$ ) is the least positive integer remainder that is obtained from the division of  $X$  by  $m_i$ . For example, consider a four-modulus system with moduli 3, 4, 5, and 7. In this system, the decimal numbers  $X = 23$  and  $Y = 14$  are represented as  $X = (2, 3, 3, 2)$  and  $Y = (2, 2, 4, 0)$ , respectively. The important feature of RNS is that fixed-point arithmetic operations can be performed on each digit individually. For example, the results of performing addition and multiplication on the above numbers are  $X + Y = (1, 1, 2, 2)$  and  $X \cdot Y = (1, 2, 2, 0)$ . These are the residue representations of the correct answers, i.e., 37 and 322, respectively. For more information on residue arithmetic, the reader is referred to Ref. 7. The fact that the digits of a residue number are independent of each other results in a number of small truth tables rather than a single large truth table. Consequently, the number of reference patterns for a particular function is significantly reduced using RNS. For example, choosing the modulus set  $M = \{4, 5, 7, 9, 11, 13\}$ , and using binary coding to represent the digits,

the number of reference patterns corresponding to full-precision addition of two 16-bit words is reduced from 36,507,189,268 to only 694.

#### C. Logical Minimization

Logical minimization techniques, such as the Karnaugh map method, the Quine-McCluskey table method, or the partitioned list method, can be used to obtain further reduction in the number of reference patterns. Minimization results for residue addition and multiplication using binary coding representation have been reported in Ref. 6.

#### D. Multi-Level Coding

Multi-level coding has been recently used as another technique for further reducing the number of reference patterns<sup>8</sup>. Here, this technique is described in detail and new results are presented.

Multi-level coding is an extension of binary coding in which more than two levels are used. For example, if three-level (ternary) coding is used, the integers zero to eight will be represented as 00, 01, 02, 10, 11, 12, 20, 21, and 22, respectively. Minimization of multi-level coded reference patterns requires a type of logic different from the commonly used binary logic. The appropriate logic, known as multiple-valued logic, is an active area of research today. Although, significant progress has been made in the theoretical aspects of multiple-valued logic<sup>9</sup>, electronic implementations of this logic have only recently begun to appear. This is partly due to the difficulties in realizing multi-state devices, and partly due to the significant progress that has been achieved in the area of binary logic systems. However, as it will be shown in the next section,

multi-level coding in optical systems can be implemented as easily as binary coding.

Minimization techniques in multiple-valued logic are somewhat different from those used in binary logic. In binary logic, if two terms in a sum-of-products logical expression are the same in all bit positions except one, they can be combined into one term which has a "don't-care" bit at that location. For example, 100 and 101 can be combined as 10X, where X represents a don't-care bit. In multiple-valued logic, terms can be combined in several ways. For example, in ternary logic, the terms 120, 121, and 122 can be reduced to 12X, where X (herein referred to as a "complete-don't-care" digit) represents a digit that can take any possible value (in this case 0, 1, and 2). If one of the above terms is absent, the other two can still be combined. For example, the terms 120 and 121 can be reduced to 12X<sub>0</sub>, where X<sub>0</sub> (herein referred to as a "partial-don't-care" digit) represents a digit with possible values of 0 and 1, but not 2.

As the number of entries in a truth table increases, the minimization procedure becomes too complex to be handled by hand. In the present work, a computer program has been developed to reduce the reference patterns for an arbitrary level coding and to obtain the minimum number of required patterns. The Quine-McCluskey technique<sup>10</sup> was extended to handle the multiple-valued logic case. In the first part of the program, a complete list of the prime implicants is obtained. Using this set, a table of choice is constructed. Then, a minimal sum set is obtained by applying the reduction rules to the table. The results for residue addition and multiplication for moduli 2 through 32 are given in Table 1. These results show that the number of reference patterns can be decreased significantly if the appropriate level of coding is used. If the modulus can be expressed



spatial locations.

### B. Recording

To implement a particular operation, all the input word combinations that produce a nonzero value at each output digit must be stored. These reference patterns are recorded as thick holograms in a photorefractive crystal, such as  $\text{LiNbO}_3$ . The recording process for each reference pattern is accomplished in three steps. Figure 3 shows the procedure for recording a reference pattern (2110). First, all the positions of the input array that are complementary to the reference pattern are made transparent and a hologram is recorded. The relative phase between the reference beam and the object beam in this recording is considered the reference phase for the other steps and it is assigned a value of zero. Then, the phase of one of the beams is shifted by  $180^\circ$  and a hologram of all positions that correspond to the reference pattern is recorded. Finally, the reference bit is recorded at a relative phase of zero. The exposure period for the first two recordings are the same, resulting in an amplitude diffraction efficiency of  $\eta_a$  for each recorded bit. The reference bit, however, is recorded with an amplitude diffraction efficiency of  $R\eta_a$ , where  $R$  is the number of nonzero digits in the reference pattern (3 for the above case).

The phasor diagram corresponding to the above example is shown in Fig. 4a. All the three steps of recording are performed with the reference beam incident at a particular angle. For other reference patterns, the position of the reference beam is changed so that each pattern is recorded with a different angle between the beams. When all the reference patterns for a particular function or operation are recorded, the recording process is complete and the processor is

as  $M = p^n$ , where  $p$  is a prime number and  $n$  is a positive integer greater than one,  $p$ -level coding is the best choice. For example, binary coding is appropriate for moduli such as 4, 8, 16, and 32, while ternary coding is beneficial for moduli such as 9 and 27. This is due to the highly regular structures of the truth tables that are produced in these cases. For a modulus that is not expressible in the above form, the proper coding level can be found among its prime factors. The prime factor that produces the largest contribution to the modulus is usually the best choice. For example, binary coding is appropriate for modulus 12 ( $= 2^2 \times 3$ ), while modulus 6 ( $= 2 \times 3$ ) benefits from ternary coding.

### III. Optical Implementation

#### A. NAND-Based Processing

The optical implementation described here is a modified version of the NAND-based processor that has been previously introduced by Guest and Gaylord<sup>4</sup>. The main advantage of this processor is its capability of operating as a parallel-input/parallel-output system. For more information on the binary-coded version of the processor the reader is referred to Ref. 4.

A schematic diagram of a ternary-coded NAND processor is shown in Fig. 2. The input array is composed of three parts: two input words and one reference bit. Each digit of the input words has two corresponding positions in the input array. If the digit has a value of "1", it is coded as a transparent aperture in the first position. Similarly, a "2" is coded as a transparent aperture in the second position. A "0" is coded as opaque apertures in both positions. In general, if  $n$ -level coding is used, each digit can have any integer value between zero and  $n-1$  and it can be positional coded with  $n-1$

ready to implement that function or operation.

#### C. Playback (Data Processing)

During the reading process, the positions of the input array that correspond to the two input words and the reference bit are made transparent, while the other locations are made opaque. The light passing through the transparent apertures upon diffraction by the holograms reconstructs the reference beams at different angles. Depending on the phase of the recorded bits, these diffracted beams are added to or subtracted from each other at the detector elements. For each output digit, if the input pattern matches one of the corresponding reference patterns, the diffracted beams cancel each other at the element of the detector array that corresponds to the matched pattern. As a result, a dark spot is produced at that element. This is detected electronically and the proper value is assigned to the corresponding output digit. If the input pattern is not similar to one of the reference patterns, the diffracted beams do not completely cancel out each other at any of the elements of the detector array that correspond to that output digit. This represents a value of "0" for that digit. Possible output values for the example studied above are shown in Fig. 4b. The numbers indicate the degeneracy of each case.

#### D. "Don't-Care" Digits

The recording process for different types of don't-care digits are presented in Fig. 5. In the case of a complete don't-care digit, the locations that correspond to that digit are made opaque during all the three steps of recording (Fig. 5a). As a result, during the reading process, the presence or absence of light at those locations has no effect on the reconstructed wavefront. Due to the positional

coding scheme that has been used to represent each digit, two types of partial-don't-care digits should be distinguished. These are: 1) partial-don't-care digits that include zero as an allowed value, and 2) partial-don't-care digits that do not include zero as an allowed value. To record a partial-don't-care digit of the first type, the locations that correspond to disallowed values of that digit are recorded at 0° phase, while the locations that correspond to the allowed values are made opaque during all the recording steps (Fig. 5b). In the case of a second type partial-don't-care digit, the locations that correspond to the disallowed values are recorded at 0° phase and those that correspond to the allowed values are recorded at 180° (Fig. 5c).

#### IV. Summary and Discussion

Optical truth-table look-up parallel data processing has been reviewed. By recording the appropriate input patterns, any function that can be described by a truth table may be implemented. This produces the advantage that all the calculations are done in advance and the results are used to construct the minimized truth table (set of reference patterns). The optical system operates as a content-addressable memory and makes a parallel comparison of the input words with the prestored reference patterns. Using the lack of angular selectivity in the direction perpendicular to the recording plane of incidence allows a parallel array of input words to be processed. Thus the processor performs as a parallel-input/parallel-output system.

The operations of addition, multiplication, and discrete matched filtering (cross-correlation) were described. The effect of coding level on the number of required reference patterns in a residue

based content-addressable memory was studied. It was found that for moduli expressible as  $M = p^n$ , where  $p$  is a prime number and  $n$  is a positive integer greater than one,  $p$ -level coding is the most efficient scheme. This is due to the significant reduction in the number of reference patterns that can be obtained by applying logical minimization techniques. In general, the prime factors that divide a modulus can be used to find the coding level that corresponds to the minimum number of reference patterns. The reduction techniques for multi-level coded reference patterns were described and the number of reference patterns required for residue addition and multiplication operations was provided for moduli 2 through 32, and for 2-, 3-, and 5-level coding.

The results presented in Table I were used to study some practical cases of addition and multiplication operations. For each case, the moduli set was selected such that it covered the required range with minimum number of reference patterns. The results are presented in Table II. The number of reference patterns in this table is equal to the number of holograms and also to the number of elements of the output photo-detector array in the optical implementation described. These numbers show that the operations studied can be implemented with state-of-the-art holography.

# REFERENCES

1. H. J. Caulfield, S. Horvitz, G. P. Tricolas, and M. A. Von Winkle, Eds., Proc. IEEE, Special Issue on Optical Computing, 72 (1984).
2. R. Bernhard, "Computers: Emphasis on Software," IEEE Spectrum 17, 32 (1980).
3. T. Kohonen, Associative Memory - A System-Theoretical Approach (Springer-Verlag, Berlin, 1977).
4. C. C. Guest and T. K. Gaylord, "Truth-Table Look-Up Optical Processing Utilizing Binary and Residue Arithmetic," Appl. Opt. 19, 1201 (1980).
5. H. J. Gallagher, T. K. Gaylord, M. G. Moharam, and C. C. Guest, "Reconstruction of Binary-Data-Page Holograms for an Arbitrary Oriented Reference Beam," Appl. Opt. 20, 300 (1981).
6. C. C. Guest, M. M. Mirsalehi, and T. K. Gaylord, "Residue Number System Truth-Table Look-Up Processing - Moduli Selection and Logical Minimization," IEEE Trans. Comput. C-33, 927 (1984).
7. M. S. Szabo and R. I. Tanaka, Residue Arithmetic and Its Applications to Computer Technology (McGraw-Hill, New York, 1967).
8. M. M. Mirsalehi and T. K. Gaylord, "Multi-Level Coded Residue-Based Content-Addressable Memory Optical Computing," in Topical Meeting on Optical Computing, Technical Digest, Sponsored by Optical Society of America, March 18-20, 1985, Incline Village, Nevada, p. 881-1.

9. D. C. Bine, Ed., Computer Science and Multiple-Valued Logic (North-Holland, Amsterdam, 1977).

10. e. g., S. Muroga, Logic Design and Switching Theory (John Wiley & Sons, New York, 1979), p. 163.

# FIGURE CAPTIONS

Fig. 1. An example of discrete matched filtering: (a) input signal,  $f(n)$ , different from the reference signal,  $g(n)$ ; (b) input signal same as the reference signal. The symbol  $\bullet$  represents cross-correlation.

Fig. 2. Schematic diagram of a ternary-coded NAND processor that implements residue multiplication modulo 6. D: detector; E: electrooptic crystal; L: Fourier transform lens; LSD: least significant digit; MSD: most significant digit; OB: object beam; RB: reference beam.

Fig. 3. Recording procedure for a ternary-coded reference pattern (2110). (a) Recording the complementary pattern (322X<sub>0</sub>) at 0° relative phase. (b) Recording the reference pattern (2110) at 180° relative phase. (c) Recording the reference bit at 0° relative phase.

Fig. 4. (a) Example phasor diagram corresponding to a recorded reference pattern (2110) in a ternary-coded NAND processor. The location of each recorded phasor at the input array is indicated with two numbers (m,n). The first number is the digit number (m=1 for the least significant digit). The second number specifies a particular position of that digit. The vertical separations between the phasors are artificially made in order to distinguish the phasors from each other. (b) Phasor diagram showing possible wavefront amplitudes at the detector element corresponding to the above reference pattern. Numbers indicate the degeneracies of the phasors.

Fig. 5. Recording procedures for patterns with don't-care digits. (a) A pattern with a complete-don't-care digit (00X1). (b) A pattern with a partial-don't-care digit of the first type (011X<sub>0</sub>). (c) A pattern with a partial-don't-care digit of the second type (1X<sub>0</sub>22). The first and third recordings are performed at 0° relative phase, while the second is performed at 180° relative phase.

Table 1. Number of Required Reference Patterns for Residue Addition and Multiplication Using Different Levels of Coding

Modulus	Addition			Multiplication		
	2-level	3-level	5-level	2-level	3-level	5-level
2	2	2	2	1	1	1
3	6	6	6	4	4	4
4	8	12	12	5	8	8
5	18	18	20	15	15	16
6	26	18	30	19	11	17
7	36	37	38	18	28	29
8	24	46	48	14	32	38
9	64	36	60	55	30	60
10	64	72	38	58	66	28
11	90	89	93	84	86	78
12	90	84	103	71	74	86
13	116	124	113	115	105	198
14	118	136	130	101	138	119
15	124	110	74	136	118	88
16	60	158	173	44	178	183
17	180	158	207	205	176	287
18	172	74	221	209	74	231
19	224	205	242	266	266	342
20	176	223	128	200	277	135
21	272	201	270	342	244	289
22	280	281	280	308	368	342
23	286	309	283	381	340	381
24	204	232	318	238	279	380
25	343	307	200	484	441	282
26	328	291	385	428	463	485
27	377	160	400	557	160	584
28	311	377	426	383	582	681
29	400	403	461	621	640	682
30	371	358	360	510	513	371
31	360	604	632	780	782	785
32	136	533	539	143	806	782

Table II Appropriate Moduli Sets and Number of Required Reference Patterns ( $N_r$ ) for Addition and Multiplication Operations.

Operation	Moduli Set	$N_r$
4-bit full-precision addition	3, 4, 6	32
8-bit full-precision addition	3, 5, 7, 8	84
12-bit full-precision addition	3, 5, 7, 8, 11	173
16-bit full-precision addition	4, 5, 7, 9, 11, 13	300
16-bit fixed-point addition	3, 5, 7, 8, 11, 13	286
32-bit fixed-point addition	5, 7, 9, 11, 13, 16, 17, 19, 23	1001
4-bit full-precision multiplication	3, 4, 5, 7	42
8-bit full-precision multiplication	5, 7, 9, 13, 16	212
12-bit full-precision multiplication	5, 7, 9, 11, 13, 17, 32	564
16-bit full-precision multiplication	5, 7, 9, 11, 13, 16, 17, 19, 23	1087
16-bit fixed-point multiplication	3, 5, 7, 8, 11, 13	234
32-bit fixed-point multiplication	5, 7, 9, 11, 13, 16, 17, 19, 23	1087

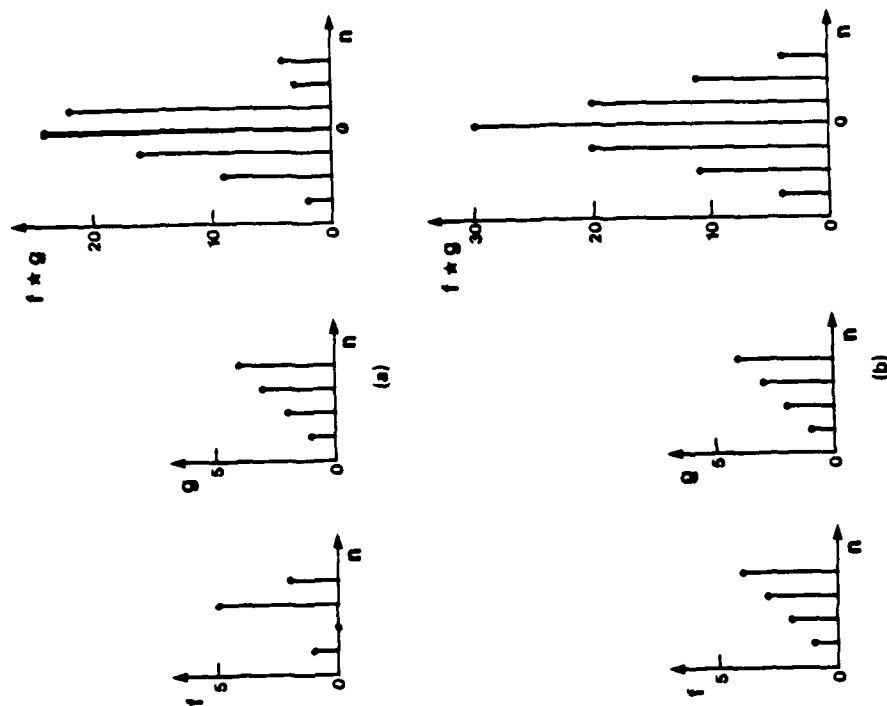
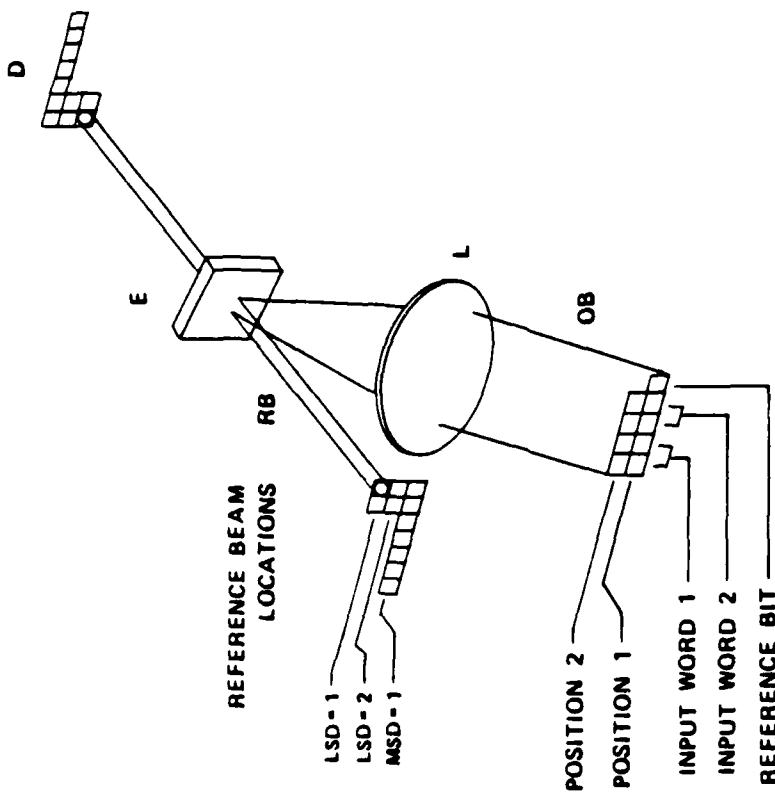


Fig. 1



11-38

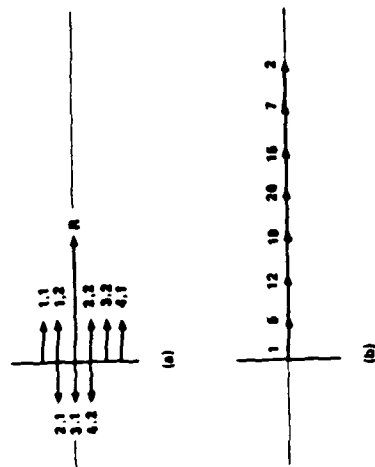
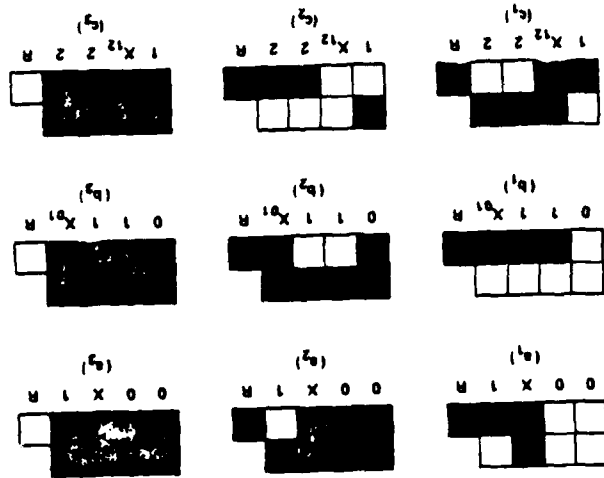


Fig. 6



## INTEGRATED OPTICAL GIVENS ROTATION DEVICE

M. M. Mirsoleni, T. K. Gaylord, and E. I. Verriest

School of Electrical Engineering

Georgia Institute of Technology

Atlanta, Georgia 30332

(Received November , 1985)

The Givens rotation operation occupies a central role in linear algebraic signal processing. A lithium niobate integrated optical coherent implementation of an elementary rotation matrix device based on thick grating diffraction to perform this operation is proposed. It is shown that existing electro-optic phase shifting and grating diffraction devices can be combined to produce a very fast Givens rotation device.

## I. Introduction

Very high speed data processing systems are needed in areas such as adaptive antenna beam forming, artificial intelligence, remote sensing, ultra-high resolution image processing, control of communication networks, air traffic control, synthetic aperture radar imaging, missile guidance, defense early warning systems, and simulation problems such as aerodynamic modeling and weather prediction requiring the solution of the Navier-Stokes equation. Real-time calculations for these highly complex and computationally intensive types of problems are largely beyond the capabilities of present day computing systems.

Monolithic integrated optical (guided wave) circuits offer the promise of very high speed processing. Devices that have been implemented in integrated optics include spectrum analyzers; analog-to-digital converters, convolvers, and correlators. In the present work, a lossless integrated optical implementation of an elementary rotation matrix ("Givens rotation" or "Jacobi rotation")<sup>1,2</sup> device that operates on optical amplitude is proposed. This device uses electro-optic grating diffraction and electro-optic phase modulator devices to achieve the required multiplication, addition, and subtraction operations in the Givens rotation.

## II. The Need for Givens Rotation

Solutions of the above listed problems can always be expressed in terms of linear algebra matrix-based algorithms.<sup>3,4</sup> The types of operations that are needed include matrix-vector multiplication, matrix-matrix multiplication, matrix inversion, solution of linear equations, solution of least square problems, singular value decomposition, the discrete Fourier transform, and calculation of eigenvalues and eigenvectors. All of these calculations may be performed by using the Givens rotation operation, repeated many times over many elements. For example, a set of linear equations of arbitrary size can be solved using the Givens rotation to triangularize the matrix of coefficients, followed by backsubstitution to determine the unknowns.

The use of arrays of integrated optical devices and fiber optics have previously been proposed to implement systolic lattice filters.<sup>5-6</sup> A form of the lattice (or ladder) filter structure, described by the square-root-normalized lattice equations, has a natural interpretation in terms of rotations. These structures are widely used for prediction and filtering in the areas of speech processing, channel equalization, seismic data interpretation, and electroencephalogram (EEG) analysis. The implementation of these filters involves a cascade of elementary sections, each consisting of a Givens rotation

and time delays.<sup>7</sup> These algorithms are also related to the Schur-Cohn stability test.

## III. Givens Rotation

The elementary rotation matrix may be expressed as

$$\begin{bmatrix} c & 0 \\ 0 & d \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}. \quad (1)$$

The flow graph corresponding to Eq. (1) is shown in Fig. 1. The Givens orthogonalization (or Givens rotation) operation is obtained when  $\sin \psi$  and  $\cos \psi$  are found such that  $d = 0$ . This operation can be used to make a particular element of a vector be zero. In fact, all but one entry of a vector can be made zero by successive Givens rotations (involving different entries of the vector). In matrix

triangularization,  $N-1$  rotations involving entries 1 and  $J$  ( $J = 2, \dots, N$ ) are applied to "zero" the first column, i.e. to transform (rotate) the first column into the vector  $[x, 0, \dots, 0]$ . The same rotations (in the same order) are used to transform columns 2 to  $N$ . The triangularization of the  $N \times N$  matrix is accomplished recursively by then zeroing the first column of the resulting lower right  $(N-1) \times (N-1)$  submatrix and so forth. Subsequent operations do not change the values in previously zeroed columns. This algorithm lends itself naturally to cascaded or pipelined hardware implementations. Due to the nonlinear  $\sin \psi$  and  $\cos \psi$  functions, the Givens rotation operation consumes a

significant amount of time and/or semiconductor material when implemented in digital electronics. This remains a problem, even though efficient bit-recursive methods using simple shift and add operations known as Coordinate Rotation Digital Computing (CORDIC) have been developed.<sup>7</sup>

#### IV. Device Configuration

The Givens rotation operation and lattice filtering simulate wave propagation phenomena. They can be modeled as lossless transmission line structures. Thus it is natural to consider wave propagation effects in constructing these devices. A coherent integrated optics implementation of an elementary rotation matrix device that operates on optical amplitude is proposed in this paper. This device uses electro-optic grating diffraction and phase shifting to achieve the required multiplications and summations in the Givens rotation operation.

The multiplications of the input amplitudes by sine and cosine are accomplished naturally and straightforwardly via diffraction by a "thick" transmission phase grating<sup>8</sup> induced by a voltage applied to periodic metallic electrodes on the surface of the device. This multiplication of amplitudes by sine and cosine using voltage-induced grating diffraction is distinctly different from the multiplication of intensities by arbitrary numbers as used in integrated optical implementations of vector subtraction, vector scalar product, and matrix-vector product.<sup>9-11</sup> In the latter applications, a voltage that is proportional to the arcsine

of the square root of the multiplier must be precalculated. In contrast to this, the present device uses the natural sine and cosine multiplication characteristics of a thick phase grating directly.

The summations in the Givens rotation are achieved by coherently combining the output waves. The phases of the waves are adjusted with electro-optic phase shifters to achieve the required addition and subtraction indicated in Eq. (1). For binary numbers, the subtraction process is analogous to the EXCLUSIVE OR operation performed with thick holograms that has been previously analyzed<sup>12</sup> and experimentally demonstrated.<sup>13</sup>

The operating principle of the device is illustrated in Fig. 2. An optical wave of amplitude "a" is incident at the first Bragg angle upon a thick grating producing a transmitted amplitude of

$$S_0 a = a \cos \psi \quad (2)$$

and a diffracted amplitude of

$$S_1 a = a \exp(j\psi_0) \sin \psi \quad (3)$$

as shown in Fig. 2a. These amplitudes are referenced to the  $x = 0$  origin at the output side of the grating. The transmitted amplitude at this point has been arbitrarily taken to be positive real. The phase factor  $\exp(j\psi_0)$  that appears in the

diffracted amplitude, Eq. (3), represents the phase difference between these waves at the  $x = 0$  output point. The sinusoidal grating refractive index may be expressed as

$$n(x) = n_0 + n_1 \cos(Kx + \phi_n) \quad (4)$$

where  $n_0$  is the average index,  $n_1$  is the amplitude of the index modulation,  $K$  is the magnitude of the grating vector ( $K = 2\pi/\lambda$ ),  $\lambda$  is the grating period, and  $\phi_n$  represents the phase of the sinusoidal grating with respect to the  $x = 0$  origin. If, for example, the  $x = 0$  origin is chosen so that  $\phi_n = 0$ , then the grating has the commonly treated sinusoidal form  $n(x) = n_0 + n_1 \cos(Kx)$ . For this case,  $\zeta_a = -\pi/2$  and the diffracted amplitude is the well known result  $S_1 = -j \sin \psi$ . In general, the phase angle of the diffracted wave is

$$\zeta_a = \phi_n - \pi/2 \quad (5)$$

Similarly, a mutually coherent optical wave of amplitude "b" is incident at the other first Bragg angle producing a transmitted amplitude of

$$S_{0b} = b \cos \psi \quad (6)$$

and a diffracted amplitude of

$$S_{1b} = b \exp(j\zeta_b) \sin \psi \quad (7)$$

as shown in Fig. 2b. The phase factor  $\exp(j\zeta_b)$  that appears in the diffracted amplitude, Eq. (6), again represents the phase difference between these two waves at the  $x = 0$  output

point. Since the "b" wave has a component in the  $-x$  direction (compared to a component in the  $+x$  direction for the "a" wave), the phase angle,  $\zeta_b$  associated with the diffracted wave is not the same as  $\zeta_a$ . For the general case, it is

$$\zeta_b = -\phi_n - \pi/2 \quad (8)$$

Coherently combining these two diffraction processes and including two external phase shifts,  $\Gamma_1$  and  $\Gamma_2$ , produces the device shown in Fig. 2c. The output amplitudes are

$$c' = a \cos \psi + b \exp[j(\Gamma_1 + \zeta_b)] \sin \psi, \quad (9)$$

and

$$d' = a \exp[j(\zeta_a + \Gamma_2)] \sin \psi + b \exp[j(\Gamma_1 + \Gamma_2)] \cos \psi. \quad (10)$$

If  $(\Gamma_1 + \zeta_b)$  is an integer multiple of  $2\pi$ , if  $(\zeta_a + \Gamma_2)$  is an odd integer multiple of  $\pi$ , and if  $(\Gamma_1 + \Gamma_2)$  is an integer multiple of  $2\pi$ , then  $c' = c$  and  $d' = d$ , the values corresponding to those given by the elementary rotation matrix operation, Eq. (1). This may be accomplished by setting the external phase shifters so that

$$\Gamma_1 = \phi_n + \pi/2, \quad (11)$$

and

$$\Gamma_2 = -\phi_n - \pi/2, \quad (12)$$

and thus  $\Gamma_2 = -\Gamma_1$ .

The "a" and "b" waves combine to form an interference pattern at the location of the grating. In the absence of any external phase shifts and for "a" and "b" both representing positive numbers, the waves are in phase and a maximum in the interference pattern is taken to be the  $x = 0$  origin. The interdigitated-electrode voltage-induced cosinusoidal grating may initially have an arbitrary phase,  $\phi_0$ , with respect to this interference pattern. Equations (11) and (12) give the required external phase shifts in order to produce the outputs "c" and "d" given by the elementary rotation matrix, Eq.(1). Some values of  $\zeta_a$ ,  $\zeta_b$ ,  $\Gamma_1$ , and  $\Gamma_2$  are given in Table I for several representative values of  $\phi_0$ . As can be seen from the table, if the grating formed by the interdigitated electrodes is a positive sine grating, then no external phase shifts are needed. This occurs if a bright fringe in the interference pattern occurs at a position in the grating where  $n(x) = n_0$  and the index is increasing in the positive  $x$  direction. This configuration gives the required addition in the "c" and the required subtraction in the "d" output. Ideally, the interdigitated electrodes should be fabricated with this relationship to the interference pattern. However, for any position of the grating electrodes, the external phase shift can be adjusted so that the correct phase relationship is established to produce the elementary rotation operation.

A schematic top-view of an integrated optical implementation of this device is shown in Fig. 3. The input light signals of amplitudes "a" and "b" are guided as TM modes in channel waveguides. The output guided wave amplitudes are "c" and "d."

#### V. Device Elements

Channel Waveguides. The optical paths are conventional single-lateral-mode channel waveguides typically about 8 microns wide that are formed in the z-cut lithium niobate by titanium in-diffusion.

Coherent Inputs. The input signals are two mutually coherent, monochromatic TM guided waves. These would probably be derived from the same laser source. The amplitudes of these waves represent the numbers "a" and "b" which individually may be positive or negative. The quantities may be expressed as  $a = |a| \exp(j\phi_a)$  and  $b = |b| \exp(j\phi_b)$  where  $\phi_a$  and  $\phi_b$  are either 0 (for a positive number) or  $\pi$  (for a negative number). If "a" and "b" are of the same sign, the waves are in phase. If the numbers have opposite signs, the waves are 180 degrees out of phase.

The use of TM guided modes (polarization perpendicular to the surface of the device) as opposed to TE guided modes allows a lower grating modulation voltage (see below) to be used. This choice also avoids possible polarization rotation effects due to the bulk photovoltaic effect in

lithium niobate because the polarization is already parallel to the optic axis.

Coherently Coupled Bends. The device shown in Fig. 3 can be constructed with straight waveguides in a simple "X" configuration. However, if curved waveguides are used, as illustrated in Fig. 3, optical power losses associated with smooth circular bends<sup>17</sup> may be avoided by using a series of straight waveguide segments of equal length for the curved portions of the channel waveguides. The bend loss oscillates as a function of segment length due to coupling between guided and unguided modes.<sup>18,19</sup> However, light coupled out of a guided mode into an unguided mode at a bend can be entirely coupled back into the guided mode at the next bend if their phase difference is an odd multiple of 180 degrees at that next bend.<sup>20</sup> This can be accomplished by correctly selecting the length of the segments. Using coherently coupled bends, a loss of 0.08 dB per 1 degree bend has been experimentally measured for titanium in-diffused channel waveguides.<sup>20,21</sup>

Etching. The grating is oriented so that the first Bragg condition<sup>2</sup> is satisfied for both input waves for the wavelength used. For an angle of incidence,  $\theta$ , the required grating period,  $\Lambda$ , is

$$\Lambda = \lambda / 2n_x \sin \theta, \quad (13)$$

where  $\lambda$  is the freespace wavelength and  $n_x$  is the principal extraordinary refractive index. For  $\lambda = 1.0$  micron,  $n_x =$

2.158, and  $\theta = 10$  degrees, the grating period would be about 1.33 microns. The Bragg regime parameters may be defined as  $\rho = \lambda^2 / \Lambda^2 n_x n_y$ , where  $n_y$  is the amplitude of the refractive index grating. If  $\rho$  is sufficiently large, the transmitted and diffracted amplitudes are proportional to sine and cosine as given by Eqs. (2), (3), (6), and (7). In grating diffraction, the parameter  $\psi$  is the grating strength parameter.<sup>2</sup> For TM guided modes,  $\psi = n_1 d_g / \lambda \cos \theta$ , where  $d_g$  is the grating thickness. The electric field component in the z direction (optic axis direction),  $E_z$ , produced by the interdigitated electrodes induces a refractive index grating whose amplitude is approximately  $n_1 n_2^2 r_{33} E_z / 2$  where  $r_{33}$  is the element of the electro-optic tensor for z-polarized light and an applied electric field in the z direction. The magnitude of the applied electric field produced by the applied grating voltage,  $V_g$ , is approximately  $2V_g / \Lambda$ . Thus the transmitted and diffracted amplitudes produced by a positive cosinusoidal grating may be expressed as

$$S_0 = a \cos(\psi d_g r_{33} V_g / \lambda \Lambda \cos \theta), \quad (14)$$

$$S_1 = -j a \sin(\psi d_g r_{33} V_g / \lambda \Lambda \cos \theta). \quad (15)$$

These amplitudes contain the desired cosine and sine multiplications. Integrated optical interdigitated-electrode electro-optic gratings for

intensity modulation and switching applications have been constructed and analyzed by numerous investigators.<sup>22-27</sup>

#### Phase Shifters. One arm of the device contains

electro-optic phase shifters to adjust the phase to produce the desired real addition at the "c" output and the real subtraction at the "d" output. These devices also utilize  $E_z$  to change the extraordinary refractive index and thus change the optical path length.<sup>28</sup> For a phase shift of  $r_1$ , the required electric field is

$$E_z = \lambda r_1 / \pi n_z^3 r_{33} L, \quad (16)$$

where  $L$  is the length of the electrodes. Since  $r_2 = -r_1$ , the phase shifter voltages will be equal in magnitude and opposite in sign.

Waveguide Crossing. Intersecting channel waveguides are capable of operating so that there is no net transfer of power from one waveguide to the other. In the present case, this is required for a grating voltage,  $V_g$ , of zero. With the addition of electro-optic modulators at the

intersections, 2x2 switches have been constructed with one set of crossing channel waveguides<sup>29,30</sup> and arrays of switches have been fabricated with multiple crossing waveguides.<sup>31,32</sup> In the absence of external modulation, the coupling of power between intersecting waveguides oscillates as a function of the angle of intersection between the channel waveguides. The fraction of the optical amplitude

in a channel waveguide that remains in that waveguide is given approximately by<sup>31</sup>  $f = \cos[(1/2) \Lambda_c \cot(\alpha/2)]$  where  $\Lambda_c$  is the coupling period and  $\alpha$  is the intersection angle of the channel waveguides. The quantity  $\Lambda_c$  is a function of the waveguide material and geometry. Thus for zero crosstalk, the intersection angle is chosen so that  $\alpha = 2\cot^{-1}(2\Lambda_c)$  where  $m$  is an integer. A high level of isolation between intersecting channel waveguides has been experimentally shown for TM modes.<sup>33,34</sup>

#### VI. Discussion

The Givens rotation is a key operation in linear algebraic signal processing. The Givens rotation device described in this work is composed entirely of electro-optic waveguide devices. It could be constructed by 1) fabricating the channel waveguides, 2) growing a buffer layer over the surface, and 3) depositing the metal electrodes. Thus it has the potential for easy fabrication. The phase shifter voltages,  $V_{s1}$  and  $V_{s2}$ , can be adjusted to give the desired performance. These voltages would then remain fixed. The grating voltage,  $V_g$ , would be initially varied until  $d = 0$  was obtained. This voltage would then remain constant while new matrix element optical inputs ("a" and "b") are applied and the corresponding matrix element optical outputs ("c" and "d") are obtained. However, greatly increased processing throughput could be obtained by having parallel arrays of these devices with the same

grating voltage,  $V_g$ , applied to all of them. This would allow simultaneous parallel computation of all revised matrix elements associated with the zero element produced.

Indeed, the Givens rotation device could be used as a fundamental building block in lattice filters, wavefront processors,<sup>26</sup> and a variety of other processing structures. In these devices the grating voltage would remain fixed and would not have to be varied. The device as schematically shown in Fig. 3 has codirectional data flow as is needed in a ladder implementation of finite impulse response (FIR) filtering. However, with minor reconfiguration as shown in Fig. 4, the device could be used with contradirectional data flow as is needed with a ladder implementation of infinite impulse response (IIR) filtering. The optical delay time associated with the device would be on the order of 25 to 50 picoseconds and thus the device could potentially function at high speeds. In lattice filtering applications, the required time delays between sections could be implemented by interconnecting the lithium niobate channel waveguides with single-mode fibers as has been done in communications applications.<sup>27,28</sup> The final output numbers may be obtained by coherent detection of the wave amplitudes and phases using optical homodyne (local oscillator frequency is the same as the signal frequency) interference techniques.<sup>29-31</sup>

The devices that must be integrated to produce the proposed Givens rotation device have previously been experimentally constructed and reported in the literature.

Although lithium niobate technology has been referenced throughout this discussion, the same rotation matrix device structure could also be constructed in integrated optical form using an electro-optic semiconductor material such as gallium arsenide. Bulk optical implementations, though not as fast, would also be possible.

This work was supported in part by a grant from the Joint Services Electronics Program and by a grant from the Strategic Defense Initiative Office administered through the Office of Naval Research.



# References

1. W. Givens, "Computation of plane unitary rotations transforming a general matrix to triangular form," SIAM J. Appl. Math 6, 26 (1958).
2. G. H. Golub and C. F. Van Loan, *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore, 1983).
3. T. Kallath, "Signal processing in the VLSI era," in *VLSI and Modern Signal Processing* (S. Y. Kung, H. J. Whitehouse, and T. Kallath, eds., Prentice-Hall, Englewood Cliffs, NJ, 1985).
4. K. Bromley, "An interview with Keith Bromley on signal processing," Optical Engineering Reports, 1 (February 1985).
5. B. Mosleh, J. W. Goodman, M. Tur, and H. J. Shaw, "Fiber-optic lattice signal processing," Proc. IEEE 72, 909 (1984).
6. M. Nazarathy and J. W. Goodman, "Systolic signal processing with integrated optical coupled-wave device arrays," in *Optical Computing Technical Digest* (Opt. Soc. Am., Washington, DC, 1985).
7. J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Electron. Comput. 8, 330 (1959).
8. H. Kogelnik, "Coupled wave theory for thick hologram gratings," Bell Syst. Tech. J. 48, 2909 (1969).

17

9. C. M. Verber, R. P. Kenan, and J. R. Busch, "Design and performance of an integrated optical digital correlator," J. Lightwave Tech. LT-1, 256 (1983).
10. C. M. Verber, "Integrated-optical approaches to numerical optical processing," Proc. IEEE 72, 942 (1984).
11. C. M. Verber, "Integrated optical architectures for matrix multiplication," Opt. Engr. 24, 19 (1985).
12. C. C. Guest and T. K. Gaylord, "Truth-table look-up optical processing utilizing binary and residue arithmetic," Appl. Opt. 19, 1201 (1980).
13. C. C. Guest, M. M. Mirsalehi, and T. K. Gaylord, "EXCLUSIVE OR processing (binary image subtraction) using thick Fourier holograms," Appl. Opt. 23, 3444 (1984).
14. E. M. Zolotov, P. G. Kazanskii, and V. A. Chernykh, Photoinduced polarization conversion in Ti:LiNbO<sub>3</sub> channel waveguides," Sov. Tech. Phys. Lett. 7, 397 (1981).
15. H. A. Haus, E. P. Ippen, A. Lattes, C. Gabriel, and F. J. Leonberger, "Double degenerate Four-Wave Mixing in LiNbO<sub>3</sub> Waveguides," Appl. Phys. B 28, 161 (1982).
16. J. F. Lam and H. M. Yen, "Dynamics of optical TE to TM mode conversion in LiNbO<sub>3</sub> channel waveguides," Appl. Phys. Lett. 45, 1172 (1984).
17. L. D. Hutcheson, I. A. White, and J. J. Burke, "Comparison of bending losses in integrated optical

18

circuits," Opt. Lett. 5, 276 (1980).

18. M. F. Taylor, "Power loss at directional change in dielectric waveguides," Appl. Opt. 13, 642 (1974).
19. M. F. Taylor, "Losses at corner bends in dielectric waveguides," Appl. Opt. 16, 711 (1977).
20. L. M. Johnson and F. J. Leonberger, "Low-loss LiNbO<sub>3</sub> waveguide bends with coherent coupling," Opt. Lett. 8, 111 (1983).
21. R. A. Becker and L. M. Johnson, "Low-loss multiple-branching circuit in Ti-indiffused LiNbO<sub>3</sub> channel waveguides," Opt. Lett. 9, 246 (1984).
22. T. K. Gaylord and M. G. Moharam, "Thin and thick gratings: Terminology clarification," Appl. Opt. 20, 3271 (1981).
23. C. M. Verber, V. E. Wood, R. P. Kenan, and M. F. Horton, "Large-angle optical switching in waveguides in LiNbO<sub>3</sub>," Ferroelect. 10, 253 (1976).
24. B. Chen and C. M. Meijer, "Bragg switch for optical channel waveguides," Appl. Phys. Lett. 33, 33 (1978).
25. R. A. Becker and M. S. C. Chang, "Electrooptical switching in thin film waveguides for a computer communication bus," Appl. Opt. 18, 3296 (1979).
26. E. M. Philipp-Rutz, R. Linares, and M. Fukuda, "Electrooptic Bragg diffraction switches in low cross-talk integrated-optics switching matrix," Appl. Opt. 21, 2189 (1982).
27. E. N. Glytsis, M. G. Moharam, and T. K. Gaylord,

"Refractive index distributions produced by interdigitated electrodes on electro-optic crystals," J. Opt. Soc. Am. A 2, xxx (1985).

28. R. C. Alferness, "Waveguide electrooptic modulators," IEEE Trans. Microwave Theo. Tech. MTT-30, 1121 (1982).
29. A. Neyer, "Electro-optic x-switch using single-mode Ti:LiNbO<sub>3</sub> channel waveguides," Electron. Lett. 19, 553 (1983).
30. L. McCaughan, "Long wavelength titanium-doped lithium niobate directional coupler optical switches and switch arrays," Opt. Engr. 24, 241 (1985).
31. E. E. Bergmann, L. McCaughan, and J. E. Watson, "Coupling of intersecting Ti:LiNbO<sub>3</sub> diffused waveguides," Appl. Opt. 23, 3000 (1984).
32. S. Y. Kung, "On supercomputing with systolic/wavefront array processors," Proc. IEEE 72, 867 (1984).
33. S. K. Korotky, G. Eisenstein, R. S. Alferness, J. J. Veselka, L. L. Buhl, G. T. Harvey, and P. H. Read, "Fully connectorized high-speed Ti:LiNbO<sub>3</sub> switch/modulator for time-division multiplexing and data encoding," J. Lightwave Technol. LT-3, 1 (1985).
34. S. K. Korotky, G. Eisenstein, A. H. Gnauck, B. L. Kasper, J. J. Veselka, R. C. Alferness, L. L. Buhl, C. A. Burrus, T. C. D. Huo, L. W. Stulz, K. C. Nelson, L. G. Cohen, R. W. Dawson, and J. C. Campbell, "4-Gb/s transmission experiment over 117 km of optical fiber

- using a  $\text{Ti:LiNbO}_3$  external modulator," J. Lightwave Technol. LI-3, 1027 (1985).
35. O. E. DeLange and A. F. Dietrich, "Optical heterodyne experiments with enclosed transmission paths," Bell Syst. Tech. J. 47, 161 (1968).
36. T. Okoshi, "Recent progress in heterodyne/coherent optical fiber communications," J. Lightwave Technol. LI-2, 341 (1984).
37. D. W. Smith, "Coherent fiber optic communications," Laser Focus/Electro-Optics 21, 92 (November 1985).

Table 1. Diffracted Wave Phase Factors ( $\zeta_a$  and  $\zeta_b$ ). External Phase Shifts ( $\Gamma_1$  and  $\Gamma_2$ ) Required to Achieve Elementary Rotation Matrix Operation.<sup>a</sup>

$n(x)$	$\phi_n$	$\zeta_a$	$\zeta_b$	$\Gamma_1$	$\Gamma_2$
$n_0 + n_1 \cos kx$	0	$-\pi/2$	$-\pi/2$	$+\pi/2$	$-\pi/2$
$n_0 - n_1 \sin kx$	$\pi/2$	0	$\pi$	$\pi$	$\pi$
$n_0 - n_1 \cos kx$	$\pi$	$+\pi/2$	$+\pi/2$	$-\pi/2$	$+\pi/2$
$n_0 + n_1 \sin kx$	$-\pi/2$	$\pi$	0	0	0

<sup>a</sup> The reference used throughout this paper, as shown in Fig. 2, is that the "a" wave has a component of its direction of propagation along the positive x axis.

# FIGURE CAPTIONS

Fig. 1. Flow graph for the elementary rotation matrix operation.

Fig. 2. (a) Transmitted and diffracted amplitudes for "a" wave. (b) Transmitted and diffracted amplitudes for "b" wave. (c) Coherent combination of "a" and "b" diffraction with external phase shifters. For properly chosen phase shifts, c' and d' become the values given by the elementary rotation matrix. All transmitted and diffracted wave amplitudes are referenced to the  $x = 0$  origin at the output side of the grating.

Fig. 3 Schematic of integrated optical elementary rotation matrix device (not to scale). The material is z-cut lithium niobate. The optical input amplitudes are "a" and "b"; the output amplitudes are "c" and "d." The grating voltage ( $V_g$ ), and phase-shifter voltages ( $V_{\pi=1}$  and  $V_{\pi=2}$ ) are shown.

Fig. 4 (a) Codirectional form of rotation matrix device for FIR lattice filtering. (b) Contradirectional form of rotation matrix device for IIR filtering.

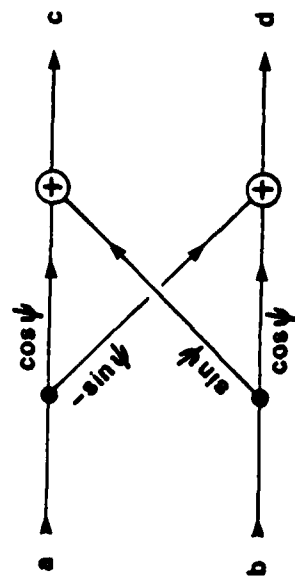


Fig. 1

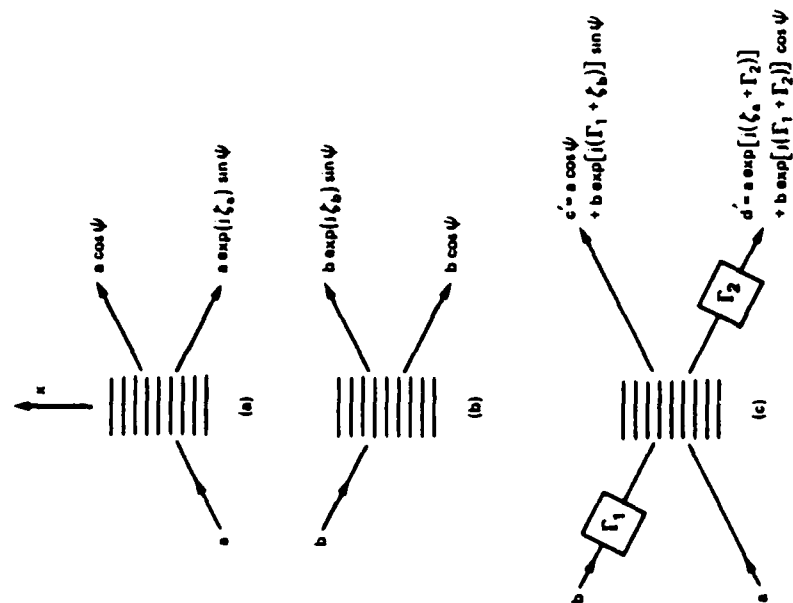


Fig. 2

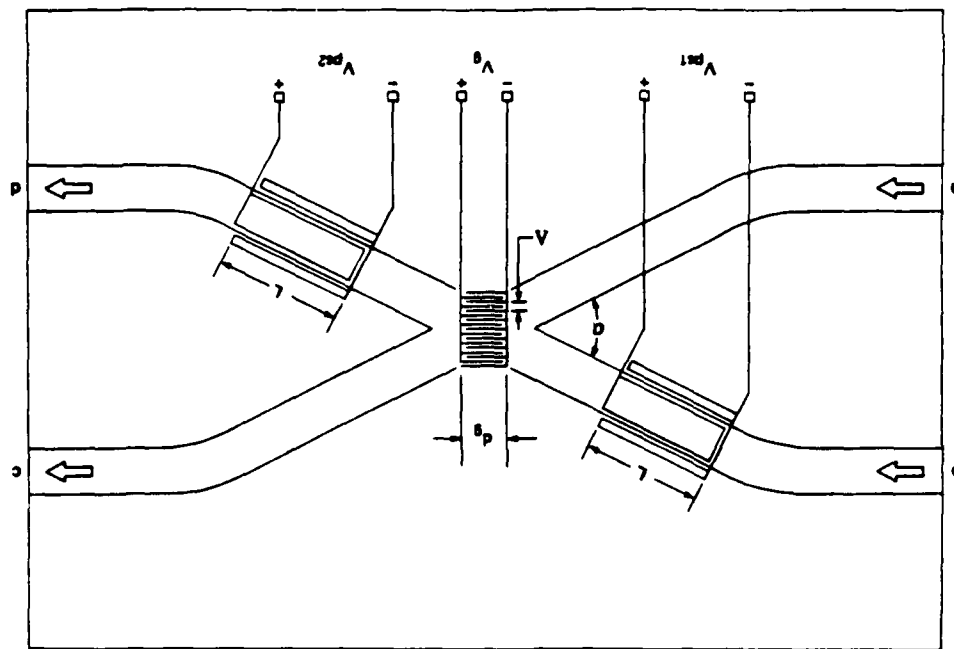


Fig. 3

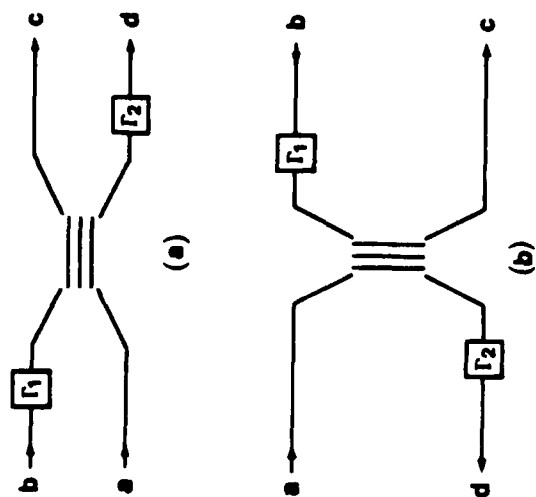


Fig. 4

**MACHINE VISION**

William Stoner, Principal Investigator

SCIENCE APPLICATIONS INTERNATIONAL CORPORATION

## SCIENCE APPLICATIONS INTERNATIONAL CORPORATION

(617) 275 2200

3 PRESTON COURT  
BEDFORD, MASSACHUSETTS 01730

William Stoner, Principal Investigator

## MACHINE VISION

## ABSTRACT

This report develops and explains work at SAIC on Machine Vision under the SDIO IS&T Optical Consortium program during the period from June to December, 1985. An architecture for a machine vision system is developed and explained in the context of Bela Julesz's findings for natural vision. This architecture contains a sensor fusion function (modeled after Julesz's Cyclopean vision), and a pattern recognition function. Optical and electronic implementations of neural nets for stereopsis and pattern recognition are presented.



## TECHNICAL SUMMARY

### 1. Objectives

Natural vision in man and animals greatly outstrips present achievements in machine vision. The objective of this effort is to tap the existing body of knowledge on natural vision, and apply this knowledge to machine vision.

### 2. Description of Work Performed and Results

The neural architecture behind stereo vision in man is examined and applied to the general machine vision problems of sensor fusion and pattern recognition. Optical processing is identified as a candidate preprocessing technique in a machine algorithm for stereo vision due to Marr.

Recent work in neural networks is examined for application to pattern recognition. The aspect and noise tolerance of neural network models, as demonstrated in the work of Kohonen and Fukushima, is identified as a potential solution to difficulties with matched filtering approaches to pattern recognition. Optical and electronic concepts to implement neural networks (either at the functional or architectural levels) are presented.

### 3. Conclusions and Recommendations

Knowledge of natural vision in humans, and related sensory systems in animals (monkeys, cats, bats) provides an extremely valuable basis for future development of machine vision. The exploitation of this knowledge will require 1) continued study of natural vision to provide additional insight and knowledge, 2) continued effort by technologists to develop parallel optical and electronic processing frameworks for machine implementation of functions similar to those found in natural vision, such as sensor fusion and pattern recognition.

Acknowledgement - We have benefited from help and discussions from several individuals:

Terry Sejnowski	JHU
John N. Lee	NRL
Dimitri Psaltis	Caltech
Ravi Athale	BDM
Harold Szu	NRL
Art Fisher	NRL

The following SAIC staff have also contributed to our thinking:

Bob Hunt  
Terry Schilke  
Devon Crowe  
Sam Rosenzweig  
John DiFrancesco

## 1.0 INTRODUCTION

This report develops and explains work at SAIC under the SDIO IS&T Optical Consortium program on Machine Vision during the period from June to December, 1985.

We use the term "machine vision" to encompass the generation and processing of three-dimensional or two-dimensional scenes from any number of diverse sensors. The sensors may detect radiation in any band from x-ray through radio wavelengths, and may be configured in arbitrary constellations in space or on-board aircraft. Combinations of active and passive sensors may be used. For example, data collected by an active inverse synthetic aperture radar (inverse SAR) and by a passive infrared (IR) telescope might be fused by the machine vision system to generate a 3-D representation of a target.

We have identified many approaches to the fusion of data from diverse sensors. Three broad categories of techniques for sensor fusion and processing can be identified:

- 1) man in loop
- 2) statistical decision theory methods
- 3) artificial intelligence (AI)

In the "man in the loop" approach, the data from the sensor(s) is prepared in graphical form for an expert to assess. If two types of sensors are involved, the imagery from each sensor might be displayed in registration (overlay). Since inverse SAR images exist in the range and one cross range dimensions, while conventional telescope images exist over the two cross range dimensions, these types of images should be combined as orthogonal projections. The human analyst brings to the data his knowledge of likely target characteristics, and his understanding of image artifacts, which might arise from glints, radar echos/wave guiding or a poor target motion solution. The obvious problem with man in the loop approaches is the difficulty in satisfying the demanding timelines required for SDI applications.

The statistical decision theory approach to vision has provided substantial progress in the area of pattern recognition. See for example the volume of reprints edited by Agrawala, Machine Recognition of Patterns. Less progress has been made in the area of sensor fusion. However, the Kalman filter technique provides a methodology for combining diverse information to refine an estimate of the state vector. For example, Wood applied Kalman filtering to the problem of image reconstruction from projections in computed tomography. (A System Theoretic Approach to Image Reconstruction, Stanford, Ph.D. thesis, 1978). In the computed tomographic application, the Kalman filter provides a natural way to "fuse" hundreds of x-ray projection measurements made at different aspect angles. This is true "tomographic" vision. Moreover, the Kalman filter also processes the measurement data in an optimal way, taking into account the known noise statistics. Unfortunately, in the x-ray tomography applications "optimal estimation" requires so much computation that it is not feasible for clinical use (Buonocore et al). Surely, the methods of statistical decision theory and optimal estimation have an important role to play in machine vision. However, if the noise statistics are distinctly non-gaussian, or the computational load of these now-classical techniques is prohibitive, a novel approach is required.

Thus the interest in artificial intelligence (AI) for machine vision. But what sort of AI? The current emphasis in AI is on expert based or rule based inference systems. Expert-based AI systems can no doubt be created which draw upon the existing knowledge of sensor and target characteristics. But the semantic domain of a rule base is very far removed from the raw image data generated by an inverse SAR or IR sensor. Rule based AI technology does not address the basic problems of pattern recognition, image fusion (as in stereo vision), or general sensor fusion. Since these visual processes are carried out at an unconscious level, we are not aware of what rules to program in a rule-based system for vision. Once the "visual" phases of the sensor fusion problem are solved, a rule-based system might be applicable for subsequent functions. However, the fact that natural vision transpires at an unconscious level suggests strongly that rule-based inference machines are poorly matched to the machine vision problem per se.

We are consequently led to an alternative approach to artificial intelligence, which draws upon what is known about natural intelligence. Fortunately, our understanding of natural vision has a respectable base in the work of Helmholtz, Wheatstone, Julesz, Hubel and Weisel, Marr, and many others.

The work of Julesz on stereo vision (Foundations of Cyclopean Perception) for example, provides valuable clues in the development of an architecture for a machine vision system capable of stereopsis. Julesz showed through his studies with random-dot stereograms that stereopsis is obtained prior to pattern recognition. Julesz demonstrated this by creating random-dot stereograms that provide no monocular clues to the pattern seen when the stereogram is fused. For the reader who is not familiar with Julesz's work we have provided, in Figure 1, an example of Julesz's random stereograms. To fuse the stereo, cross your eyes so that the left and right random dot patterns are superimposed. If you succeed in obtaining stereopsis, you will see a pattern in the stereo view which is not evident under monocular viewing.

We have generalized this result for the case of multiple sensor fusion in Figure 2, which presents a general architecture for 3-D scene generation and pattern recognition that is consistent with Julesz's findings for stereopsis. The machine vision program which we have mapped out under this initial SDIO IS&T effort encompasses both the further development of a machine vision architecture, and the development of the individual functions identified in this architecture, e.g., machine sensor fusion, and pattern recognition.

An indication of the potential for machine sensor fusion is provided by the work of Marr (Vision) on stereopsis. Our goal is to find more general solutions for trinocular vision as well as binocular vision. Multiple sensor fusion to clear up ambiguities present in ordinary stereo vision of complex 3-D scenes might be called "tomographic vision." This is illustrated in Figures 3 and 4. The term tomography is used in the field of x-ray imaging to identify various multiple source geometry techniques

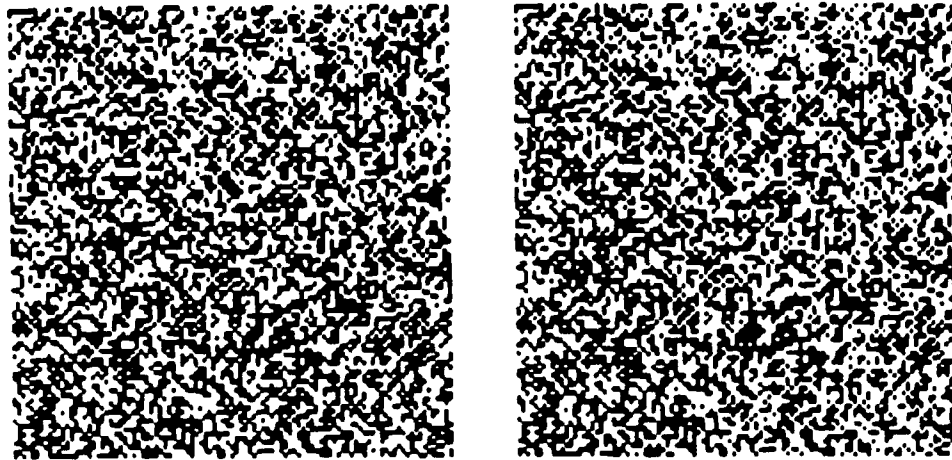


Figure 1. A Random-dot stereogram which when monocularly viewed appears as an aggregate of random dots. However, when stereoscopically fused a diamond is perceived hovering over the random background.  
(Taken from Bela Julesz, Foundations of Cyclopean Perception, p xi.)

MODULAR APPROACH TO MACHINE VISION

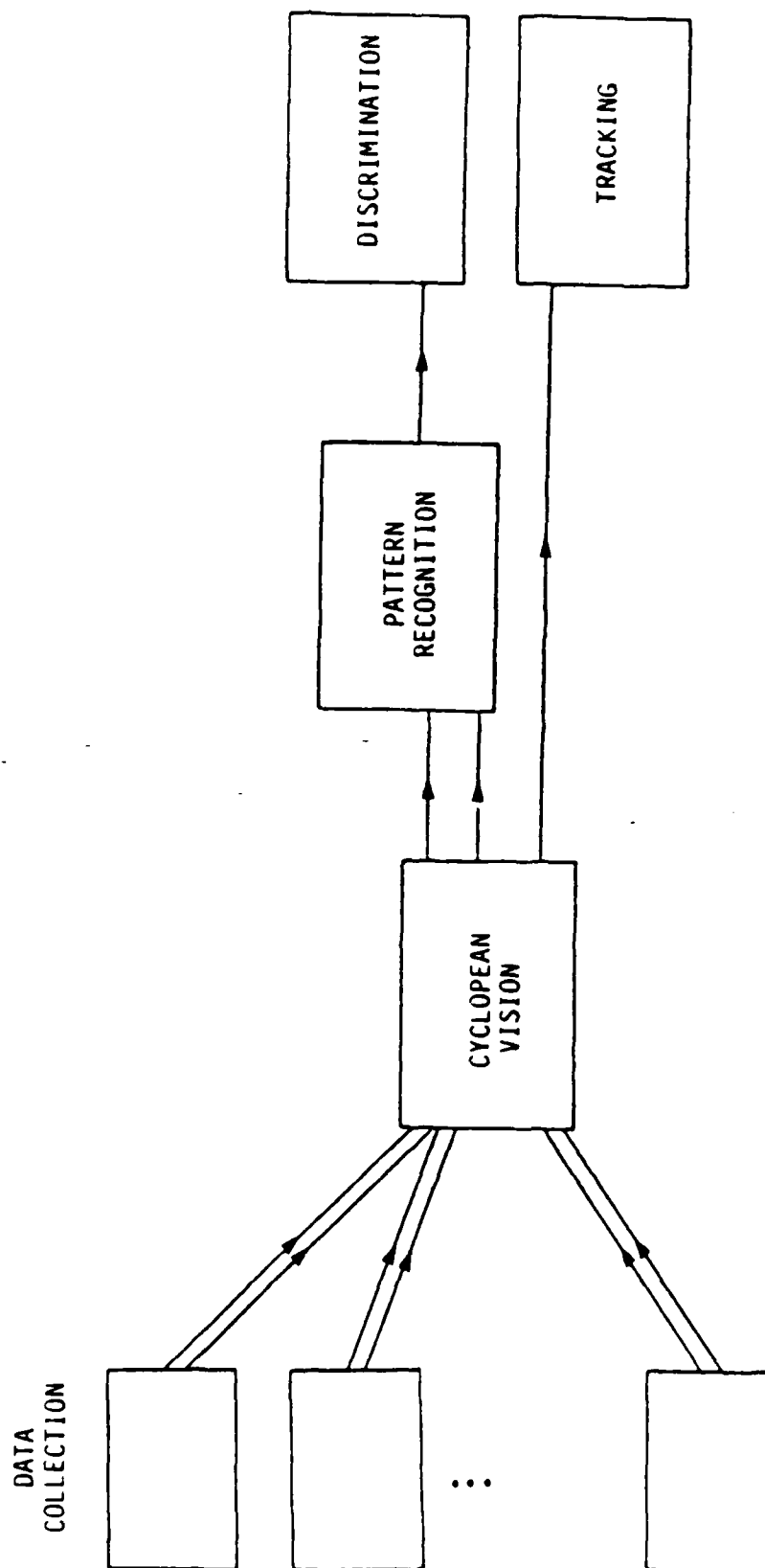


Figure 2. Architecture for multiple sensor fusion and image exploitation.

FALSE TARGET PROBLEM WITH BINOCULAR STEREO

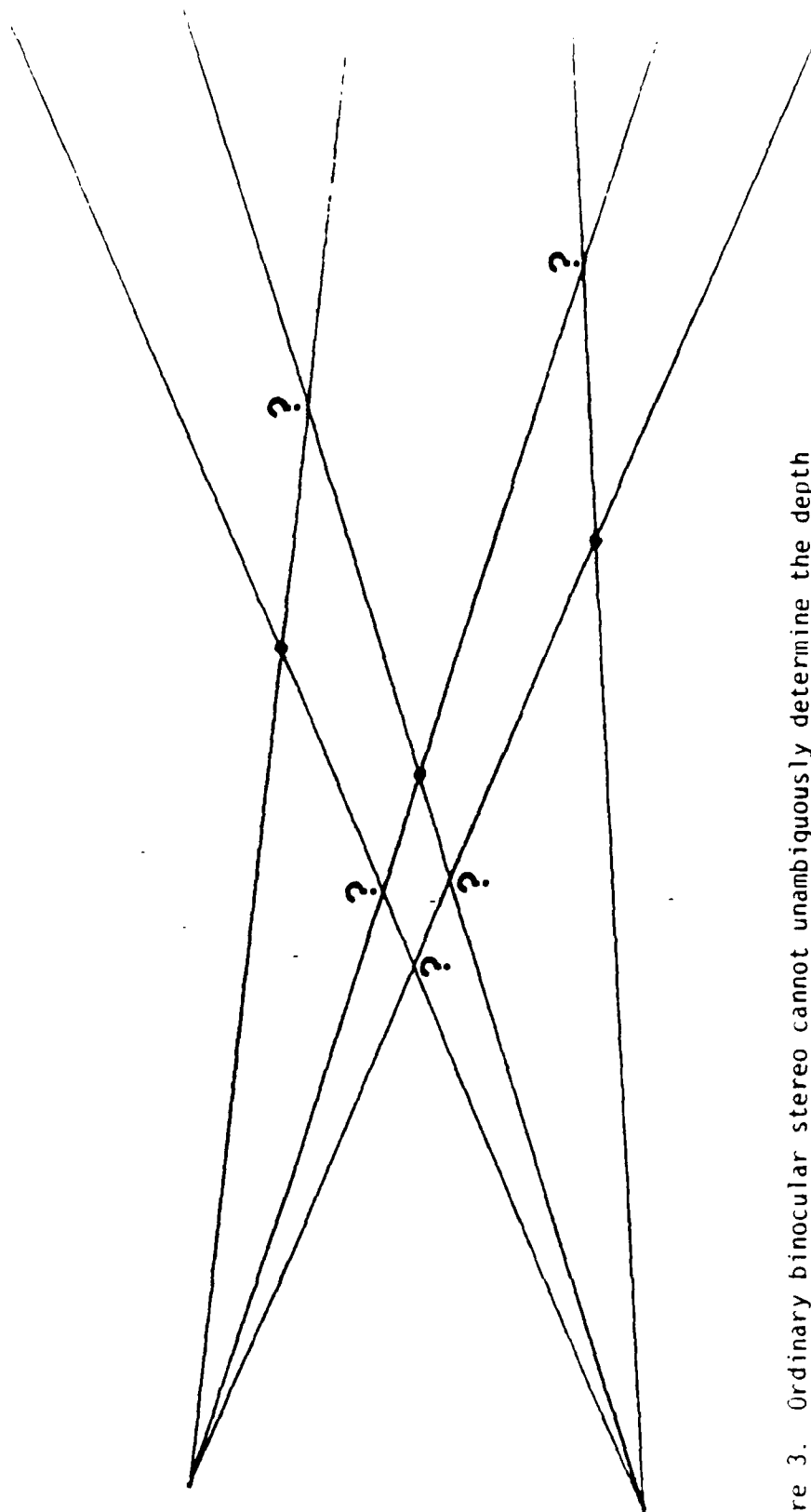


Figure 3. Ordinary binocular stereo cannot unambiguously determine the depth of multiple unresolved points in space.



MULTIPLE VIEWS REDUCE OR ELIMINATE FALSE TARGETS

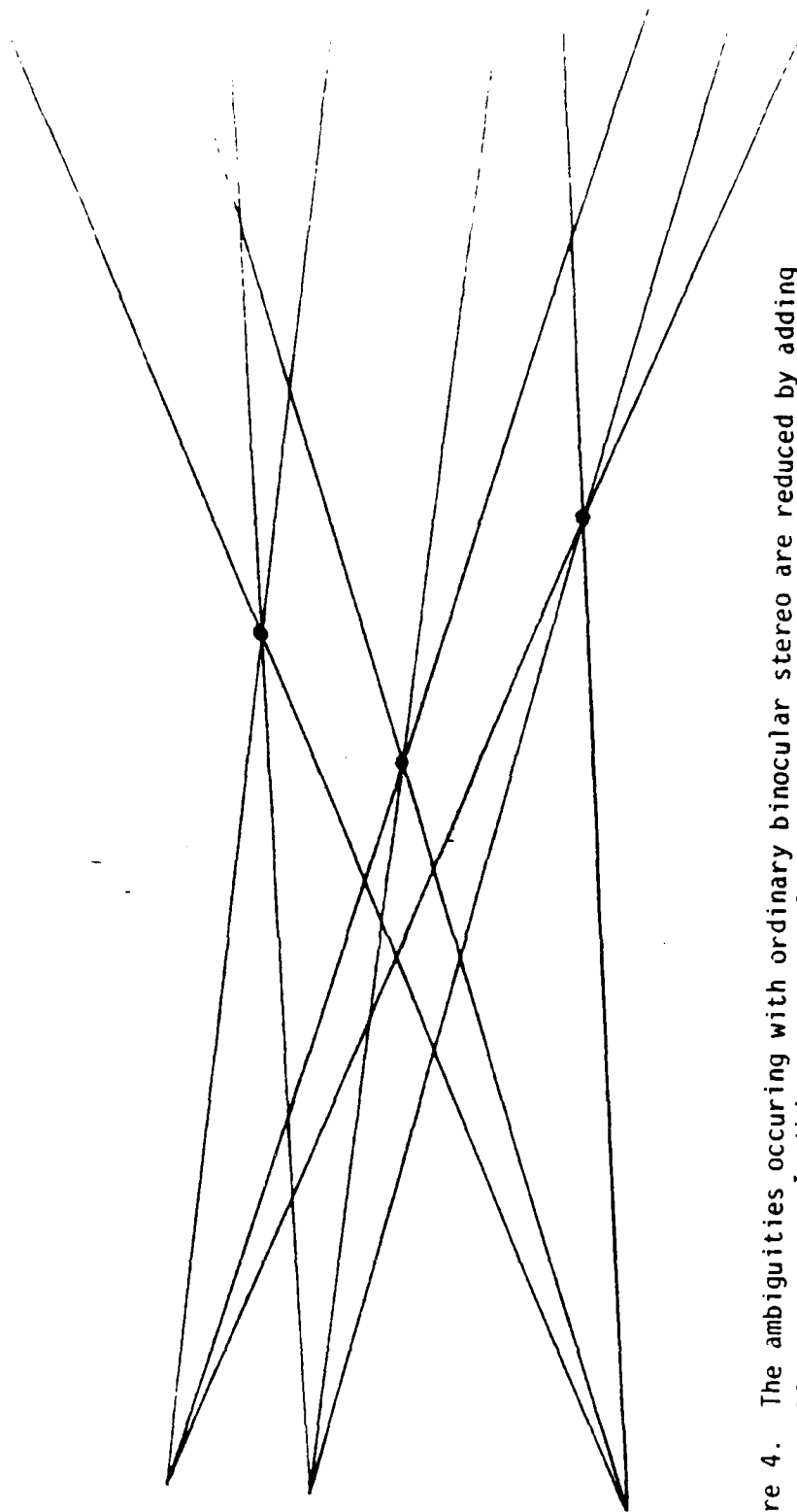


Figure 4. The ambiguities occurring with ordinary binocular stereo are reduced by adding more sensors. In this case a total of three sensors provide unambiguous depth for three targets in space, because only these target locations are consistent for all three sensors.

for obtaining 3-D image structure. (For a discussion of tomography see B. G. Ziedes des Plantes, "Body-section radiography: History, image information, various techniques, and results," *Australas, Radiol.*, Vol 15, pp 57-64 1971, or books by M. Ter Poggosian or H. H. Barrett on medical x-ray imaging).

Applicable work on pattern recognition for machine vision has been done by the optical processing community for many years. Recently, an alternative approach based on neural networks has also shown great promise. Beginning with embryonic ideas in the United States on the "perceptron" (F. Rosenblatt, Principles of Neurodynamics), workers in the neural network field have been seeking to understand how neural networks can perform pattern recognition. Fukushima in Japan has had some success with a neural network model he has developed under the name "neocognitron."

Another approach for pattern recognition using neural networks is found in the associative memory work of Kohonen. (Self-Organization and Associative Memory).

Both the work of Fukushima and of Kohonen shows advantages over traditional matched filtering for pattern recognition. The difficulty with matched filtering is its intolerance to object distortion or aspect changes. Fukushima and Kohonen have demonstrated neural network models which can adapt to such changes.

## 2.0 MACHINE VISION AND NATURAL VISION

The performance of natural vision in man and in many animal species exceeds the capability and know-how of present machine vision technology. Our curiosity is tweaked; what are the processes, forms and functions of natural vision systems? Specific high level capabilities of natural vision that are of relevance to machine vision include:

- detection of a given class of objects in the presence of a disturbed or cluttered background
- stereo vision
- sensory fusion, e.g., cuing of vision by hearing
- reliable performance with noisy components (neurons)

Low level features of natural vision that are of interest include:

- retinal readout and pre-processing mechanisms
- dynamic range accommodation
- flat fielding to achieve uniform response from non-uniform detectors

In the quest towards machine vision systems that can out-perform natural systems, knowledge of natural vision can be exploited in several ways. Just the existence of human vision capabilities, packaged compactly, and powered efficiently demonstrates what may be achieved, and motivates our search for more knowledge about vision.

In the case of radar vision, the existence proof is provided by bats. The sound-locating apparatus of bats weighs only a fraction of a gram, yet can do the job of radar equipment weighing hundreds of kilograms. Among the sophisticated functions performed by bats are ranging and near optimal pursuit of prey (mosquitos, moths); Moving Target Indication (MTI), target discrimination, accurate direction finding, and non-interfering operation in the presence of a great number of other bats (Radar Made Easy, by M. Razmakhnin). Behavioral experiments show that bats can "hear" surface roughness introduced by 20-50 micrometer deep scratches. Remarkably, laboratory investigations of neuron responses in the bat inferior colliculus (one of the way-stations for nerve impulses from the cochlea to the auditory cortex) have shown the presence of neurons that can detect a 15 microsec time lag in the arrival of a sound pulse at the left and right ears (Gerhard Neuweiler, "How bats detect flying insects ").

Knowledge of machine vision can assist in our interpretation of natural vision, and vice-versa. As the principles underlying natural vision are gradually revealed, machine vision workers can try to initiate these principles. The actual implementations need not, however, be direct imitations; once the processes are understood, the best implementation is likely to use strategies and devices that have eluded biological evolution.

For example, present work in artificial intelligence emphasizes expert based (knowledge based) systems that operate in specialized subjects. The rules upon which such systems operate are combined logically in "inference machines" providing chains of deductions that mimic the expert's reasoning. This approach to artificial intelligence (AI) has proven useful for certain applications. But the partitioning of the AI system into a "rule base" and an "inference machine" may be a misleading paradigm for actual human thought. In fact, there is evidence from work on associative memories that the "rules" humans seem to apply in many situations may actually emerge from the accumulative associations of stimuli patterns in an associative memory (Anderson and Hinton, p 21 Parallel Models of Associative Memory).

Since humans carry out most visual processes at an unconscious level, it is likely that the neural architecture supporting vision is very well adapted to this task, and provides a good architectural basis for machine systems to imitate. On the other hand human capability for logical or deductive processes is not impressive compared with present day computer capabilities, and consequently the neural architecture supporting conscious thought may not be a good candidate for imitation. In summary, we find motivation for the comparative study of natural and machine vision systems because of the:

- 1) existence proofs provided by human vision capabilities
- 2) complementary understanding derived by knowledge of both machine and natural vision systems
- 3) possibility of imitation, either at the algorithm or architectural level.

### 3.0 SENSOR FUSION

Natural stereo vision is the most familiar example of sensor fusion by a neural network. Other examples are precise direction finding in bats through binaural hearing, and in cats, of registration in the visual cortex of neurons mapping visual space with neurons mapping auditory space. (G. Neuweiler, "How bats detect flying insects,"; also, G. Hinton and J. Anderson, Parallel Models of Associative Memory, p39).

Algorithms for binocular stereo vision have been developed by Marr and his colleagues, and are presented in Marr's book Vision. We shall concern ourselves here with the "zero-crossing" algorithm, which is applicable for resolved surfaces. The extraction of depth information from a pair of stereo images involves simple geometry once corresponding features in the left and right images are identified. The problem Marr solved was to find an automatic way in which corresponding features could be paired up, without making errors in the presence of closely spaced, similar looking features. Marr's algorithm combines two strategies. One strategy is to look for edge-like features, and to ignore grey-level (image intensity) structure. The edge-like features are brought out by applying a second-order differential filter to the images. Sensitivity to edges without regard to direction is achieved by using a symmetrical differential operator, the Laplacian, denoted by  $\nabla^2$ . The other strategy is to conduct the search for corresponding features with a very blurry form of the images first, and after this step has resolved the rough depth structure, to look for more corresponding features with successively less and less blurred images, until the full resolution is reached.

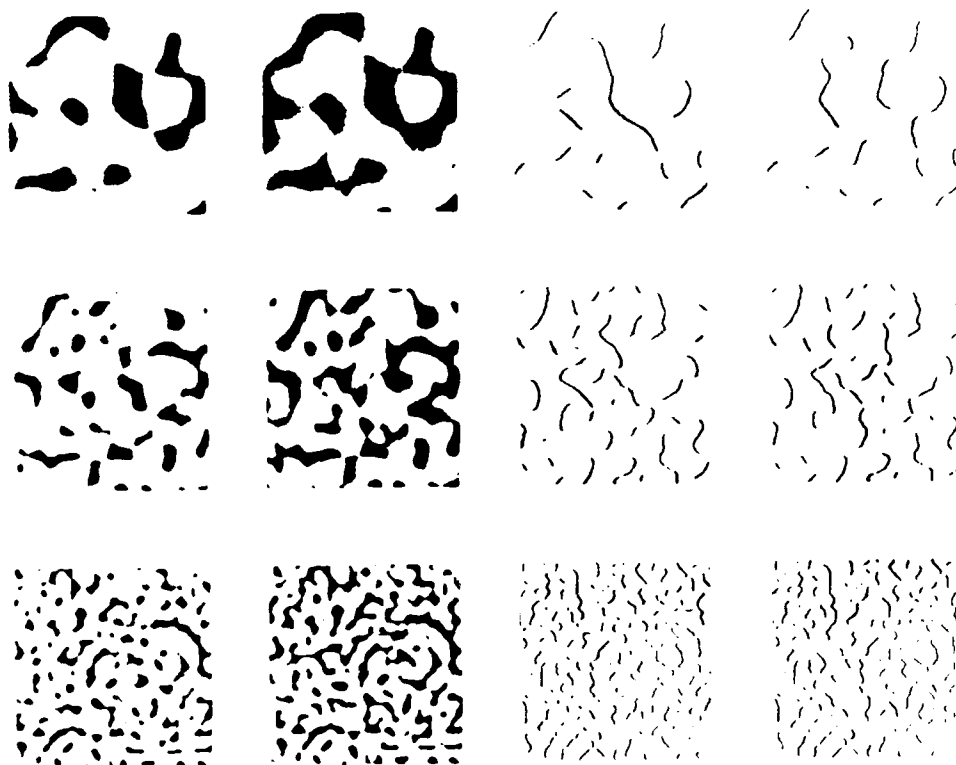
By working in stages, the density of confusing features to the pair-up is kept low enough at each stage to avoid mispairing of features in the left and right images. This procedure is illustrated in Figure 5.

An optical implementation of this algorithm seems possible, since the blurring and zero-crossing detection are linear operations that can be implemented with coherent or incoherent optical processors (Goodman,

(1)



(2)



(3)

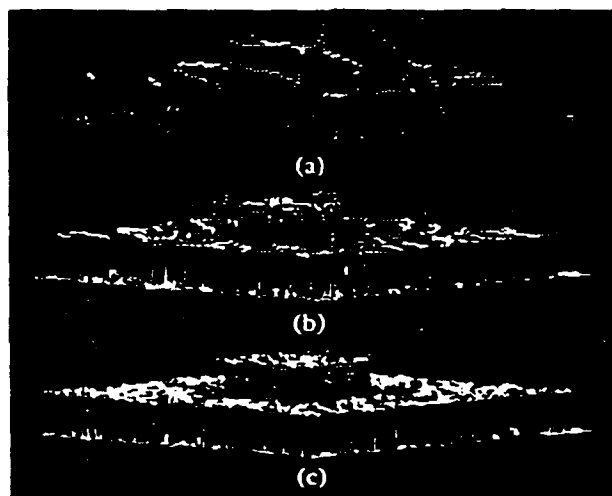


Figure 5. In (1) is shown a stereo pair. In (2) blurred versions of this stereo pair are shown on the left; the zero crossings of these blurred images are shown on the right. The successive results of stereopsis at increasing resolution is shown in (3), a through c. (Taken from Marr, Vision)

Introduction to Fourier Optics; Stoner, "Incoherent optical processing via spatially offset pupil masks," Applied Optics 17, p 2454 1978).

A diagram of the zero-crossing stereopsis algorithm is shown in Figure 6. With the possible exception of the final depth disparity computations, the operations can be implemented optically: gaussian blurring, zero-crossing detection with the Laplacian, and cross-correlation of right and left zero-crossing images to determine depth disparity.

#### 4.0 IMPLEMENTATION OF NEURAL NETWORKS FOR PATTERN RECOGNITION

Implementation of neural nets can be made at the functional level or at the architectural level. The function of a neural net might be equivalent to iterative matrix vector multiplication and thresholding. Various electronic or optical matrix-vector multipliers might be used, in either analog or digital implementations. Since analog optical matrix vector multipliers have been developed at a number of laboratories (NOSC, Stanford, Carnegie-Mellon, Caltech) we shall not delve into this approach here. Rather, we shall look into the possibility of architectural level implementations of neural net concepts. One such architecture supports associative memories which can perform pattern recognition with tolerance to aspect angle and noise. These properties are likely to be important to target identification and discrimination in a SDI system. See Figures 7 and 8.

An associative memory may be constructed in a simple way following Steinbuch (see Kohonen, Self-Organization and Associative Memory, p 73). The Steinbuch Learning Matrix (Figure 9) is a system of crossing signal paths, with an adaptive connection at each crossing. The learning mechanism is extremely simple. During a training session the proper output response is applied externally to the b-lines for each of a set of representative input stimuli applied to the e-lines. Those connections between the input and output which are simultaneously active (positively correlated) during the training session are strengthened. That's all there is to it. This learning mechanism of strengthening the connections that

# ZERO-CROSSING STEREOPSIS ALGORITHM

MARR, POGGIO (1979)

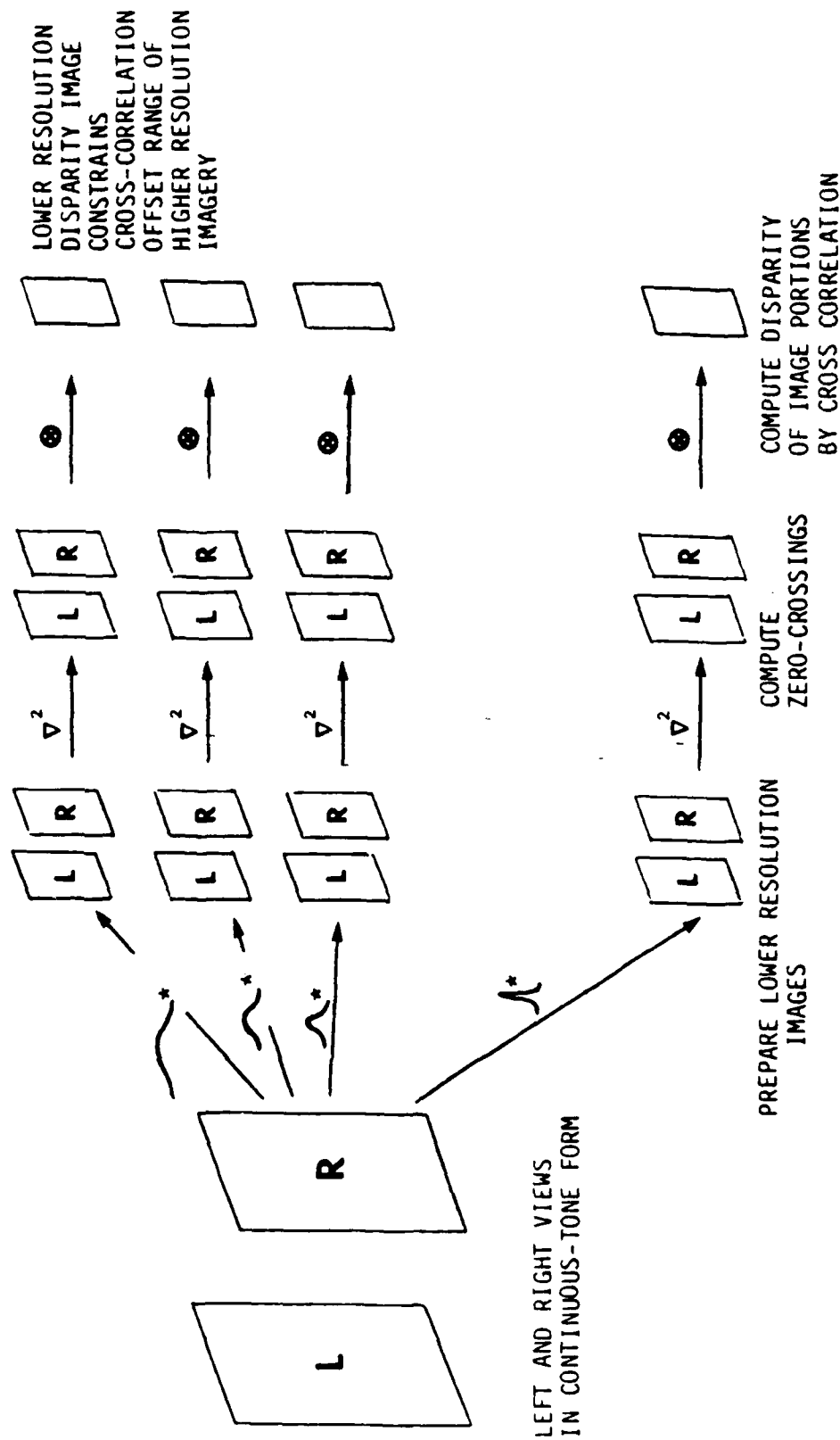


Figure 6. Diagram of the operations for Marr's zero-crossing stereopsis algorithm.



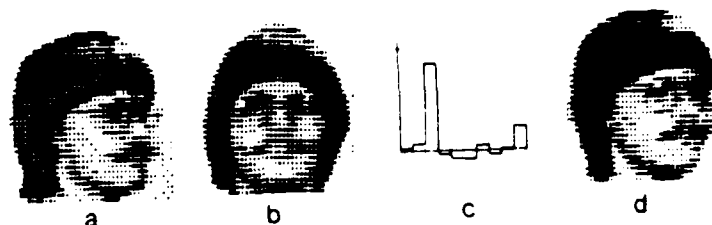


Figure 7. Demonstration of associative recall with tolerance to aspect change. Ten classes of pictures were stored. Within each class, the pictures are images of a face taken at 5 different aspect angles. Part (d) shows a test image of the person in (a) and (b), taken from an angle not originally stored. Part (c) shows correct classification of (d) by the associative memory.  
 Taken from Kohonen, Lehtio and Oja, "Distributed Associative Memory," in Parallel Models of Associative Memory, edited by G. Hinton and J. Anderson.

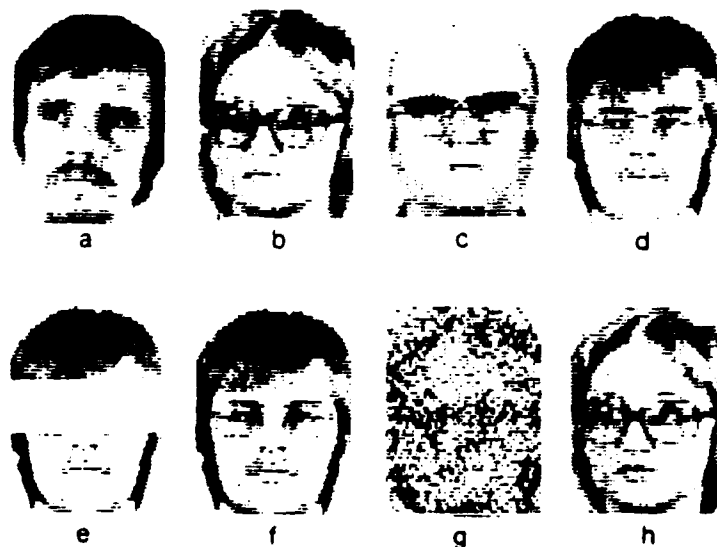


Figure 8. Demonstration of autoassociative recall with incomplete or noisy inputs. Parts (a) through (d) show 4 of 100 stored images. Parts (e) and (b) show an incomplete input which is correctly associated, and parts (g) and (h) show a noisy input which is correctly associated. (Taken from Kohonen, Lehtio and Oja, "Distributed Associative Memory," in Parallel Models of Associative Memory, edited by G. Hinton and J. Anderson.

## The Learning Matrix

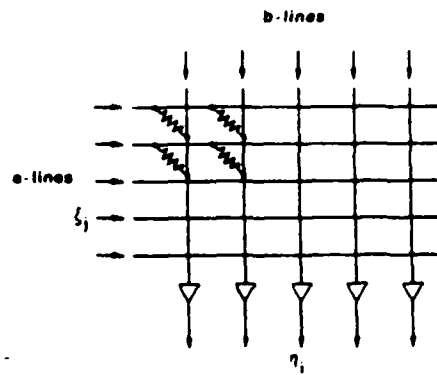


Figure 9. The Learning Matrix. Inputs come in the e-lines, outputs along the b-lines. The connections (shown in four places as resistors) must be adaptable for learning to occur. Training is performed by forcing the b-lines in conjunction with stimulus along the e-lines, with Hebbian adaptation of the connections. Thresholding on the output lines is indicated by the triangular symbols. Figure taken from Kohonen, Self-Organization and Associative Memory.

are positively correlated is "Hebbian" since Hebb hypothesized that biological neural networks learn by strengthening those synapses which connect concurrently active neurons.

It has proven very difficult to directly study synapse growth. However, there is evidence that nerve growth is stimulated by electrical current and by the concentration of "nerve growth" chemicals (Borgens et al, "Enhanced Spinal Cord Regeneration in Lamprey by Applied Electric Fields," Science 213 p 611 1981 and Barnes "What Makes Nerves Regenerate," Science 230 p 1024, 1985).

The practical question is: What sort of mechanisms might we invent or develop to realize Hebbian connections in an optical or electronic neural network? Since we require positive correlation in the activity on either side of a synaptic connection, we are looking for a synapse weight factor that grows as  $\langle xy \rangle$  where  $x$  is the activity on one side of the synapse,  $y$  is the activity on the other side, and the brackets  $\langle \rangle$  indicate averaging over many stimulus response cycles.

(Of course there are other possible candidate growth factor relationships, such as  $\langle f(x) g(y) \rangle$  where  $f(x)$  and  $g(y)$  are monotonic functions, but the  $xy$  term would enter into the Taylor series expansion of  $f(x) g(y)$ , and so we are capturing the essential factor if we restrict ourselves to  $\langle xy \rangle$ .)

Holography provides a candidate implementation of Hebbian learning, because the formation of a hologram depends on the product  $xy$  in the interference between two beams; call them the stimulus beam  $x$  and the response beam  $y$ .

Four-wave mixing materials such as BSO and barium titanate might be utilized for real-time, adaptive holographic associate memories. See the set-up in Figure 10 (J. Huignard et. al., "Phase-conjugate wavefront generation via real-time holography in  $\text{Bi}_{12}\text{SiO}_{20}$  crystals,"). A nonlinear, iterative associative memory may be implemented by placing the linear holographic memory in a laser cavity, as shown in Figure 11. For a related concept, see D. Pepper, Scientific American, Jan 1986, p 79.

coherent illumination from an argon laser

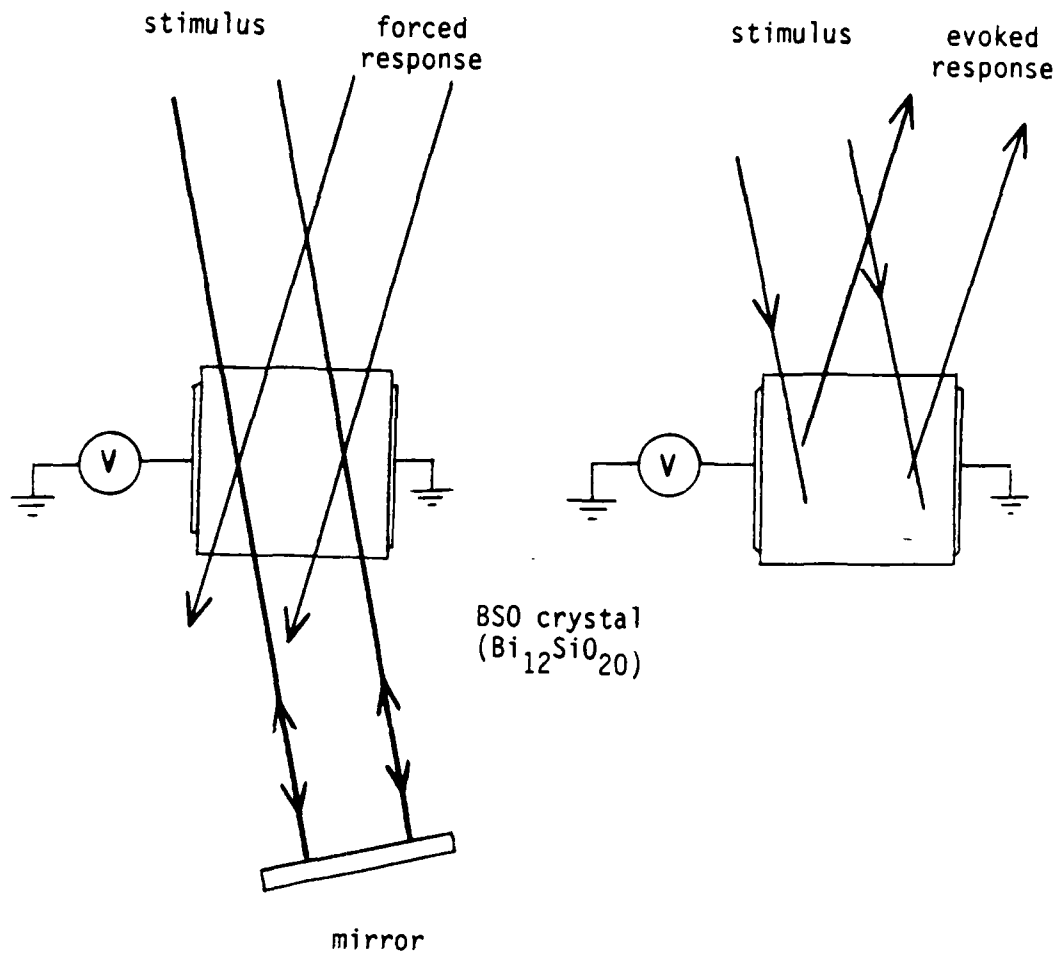


Figure 10. Application of real-time holography to form a linear associative memory

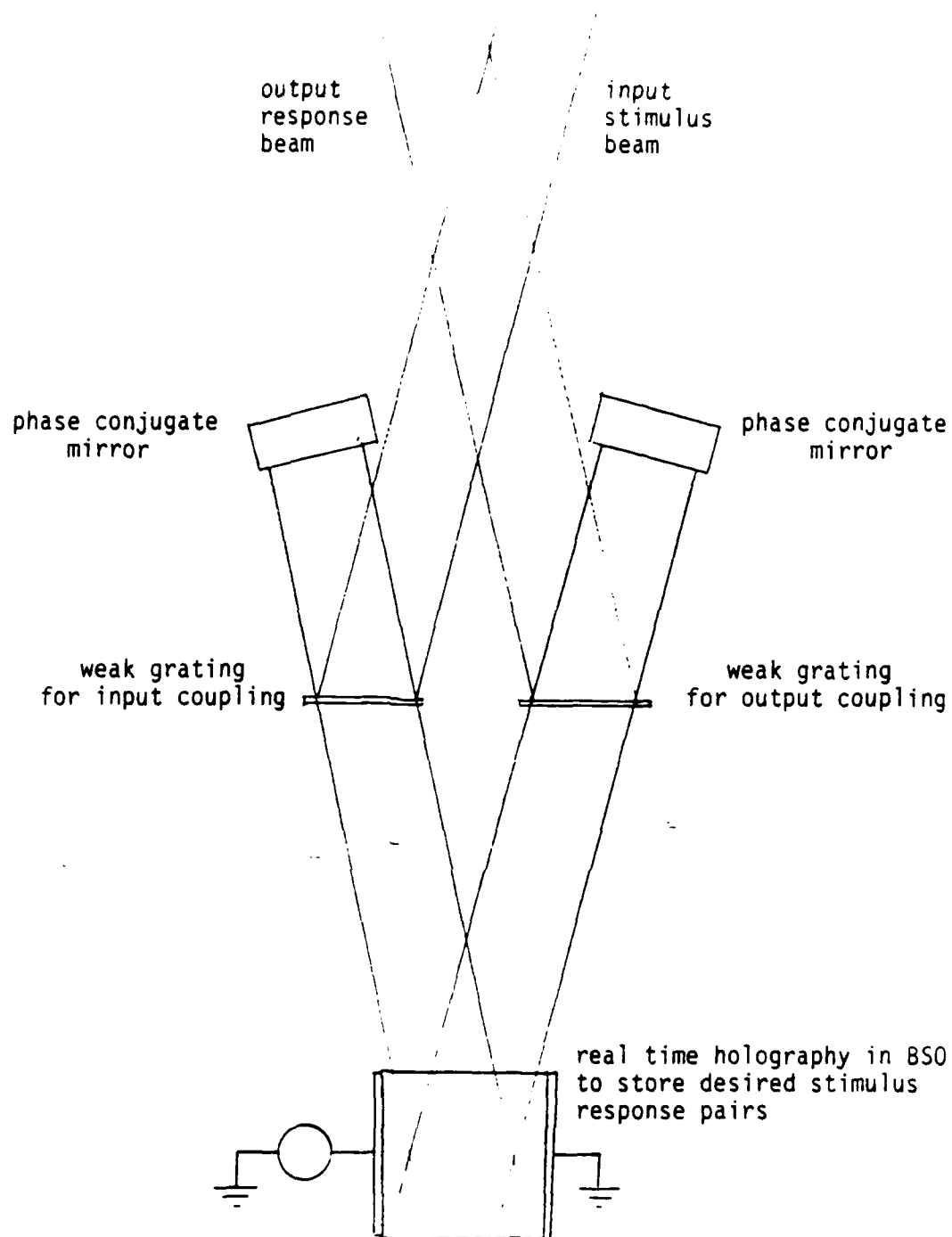


Figure 11. Nonlinear associative memory concept. Stimulus/response pairs are stored as holograms in the BSO crystal. The phase conjugate mirrors and the holograms constitute a laser resonator, with the phase conjugate mirrors providing system gain. Mode selection is triggered by the input stimulus, and the system settles into the stimulus/response mode which is closest to the input stimulus, as this mode has the highest local maximum in system gain.

In analogy to the formation of rock connections in caves between stalagmites and stalagmites, we can imagine connections growing between the input and output paths of a Learning Matrix (Steinbuch). Kohonen mentions the memistor as one such automatic connection forming device. Another (more attractive) possibility mentioned by Kohonen is the use of FET devices as variable resistors in a Learning Matrix. We point out that programming of the Learning Matrix could then be achieved optically, using an optically addressed FET gate. Here we seek to exploit the high sensitivity of the FET to small changes in gate voltage, and the attractive feature of low cross-talk between the optical addressing signals and the electronic pulses used to operate the Learning Matrix. A slightly different idea is to use a slowly responding photoconductor at connecting nodes. After a programming exposure with an appropriate visible light pattern, a photoconductor such as cadmium sulfide will retain enhanced conductivity for many electronic readout cycles before needing to be refreshed. Of course, this refresh exposure can be used for re-programming the previous connection strengths. (See Figure 12).

An all optical implementation can be based upon the principle noted earlier, for implementation of Hebbian synapse weights through a growth mechanism proportional to the average product  $\langle xy \rangle$ . This time, instead of a holographic principle, we use a photo-bleaching effect which requires light of two different wavelengths to proceed:  $\lambda_a$  and  $\lambda_b$ . One possibility is an optical pumping system, such as that diagramed in Figure 13. The initial optical connection node contains material in a metastable, absorptive state. Exposure to light of  $\lambda_a$  and  $\lambda_b$  pumps the material to a state which decays to a transparent state.

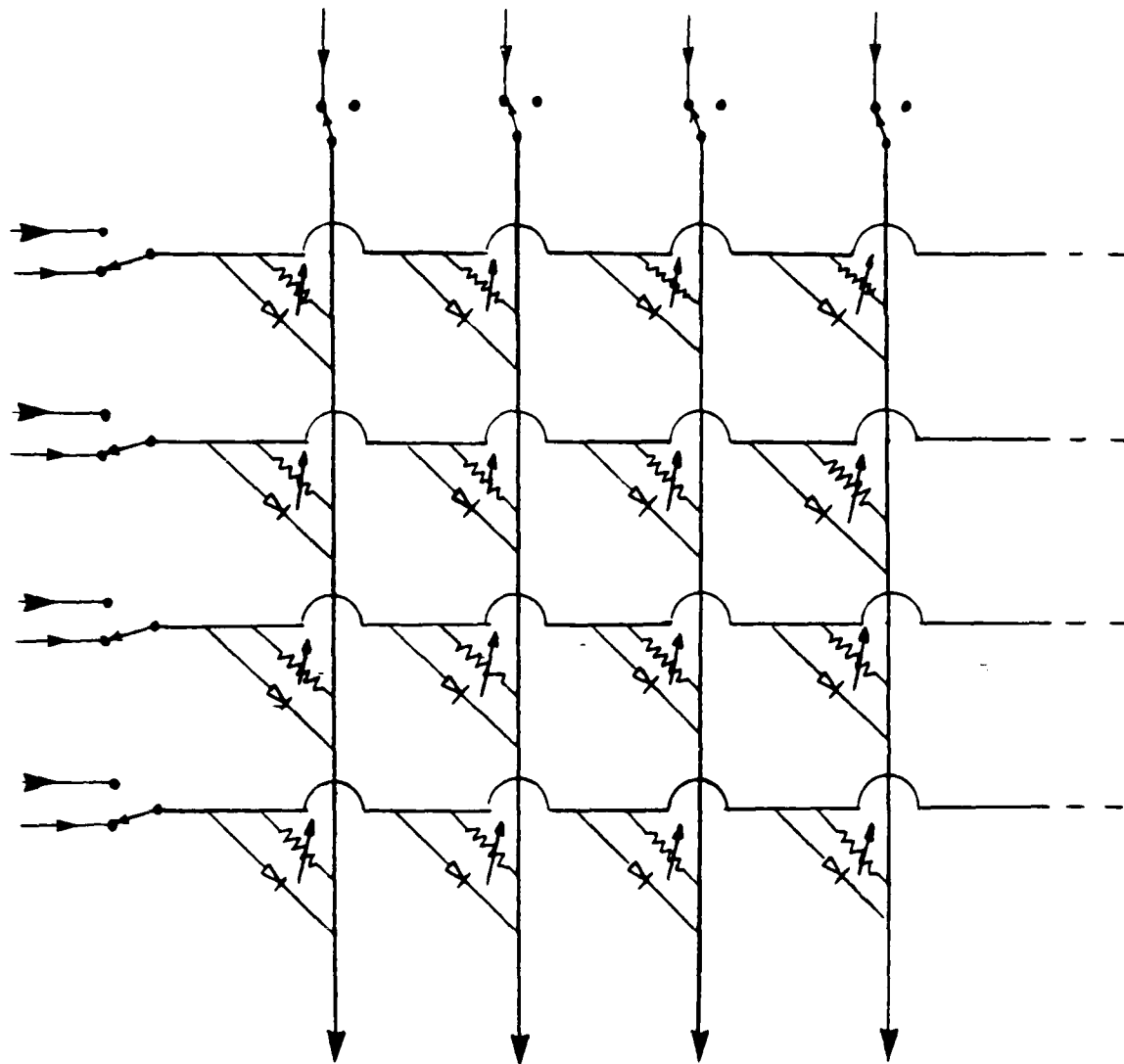


Figure 12. Learning Matrix implementation concept. The matrix connections shown as variable resistors are made out of a slowly responding photoconductor like cadmium sulfide. The photoconductive connections are "set" by visible light exposure from proximity coupled LEDs, during a training session. For the training session the switches are as shown, and the LEDs are forward biased by a combination of input simuli and forced response potentials. The switches are reversed for operation, leaving the LEDs back-biased.



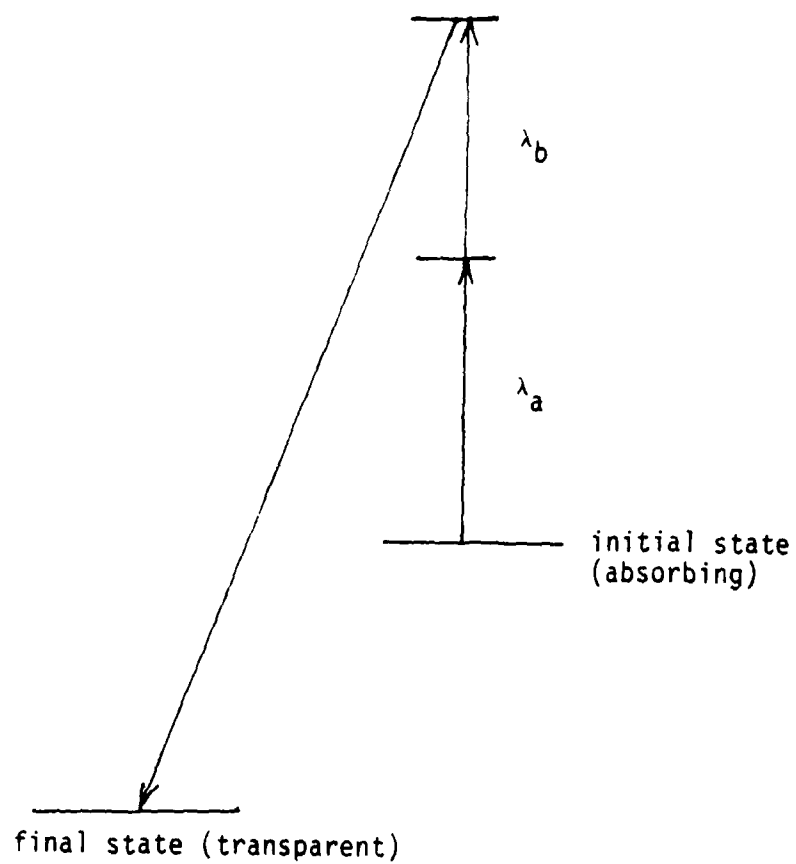


Figure 13. Optical pumping scheme requiring radiation at  $\lambda_a$  and  $\lambda_b$  to switch media from absorbing to transparent state.

## BIBLIOGRAPHY

1. A. Agrawala, Editor Machine Recognition of Patterns, (IEEE press 1977).
2. M. Buonocore, W. Brody, A. Macovski, "A Natural Pixel Decomposition for Two-Dimensional Image Reconstruction," (IEEE Trans. Biomed Eng. BME-28 Feb 1981).
3. K. Fukushima, S. Miyake and T. Ito, "Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition," IEEE Trans. on Systems, Man and Cybernetics, SMC-13, Sept/Oct 1983 p 826.
4. J. Goodman, Introduction to Fourier Optics, (McGraw-Hill, New York, 1968).
5. G. Hinton and J. Anderson Parallel Models of Associative Memory (Lawrence Erlbaum Associates, publishers, Hillsdale, New Jersey, 1981).
6. J. Huignard et. al., "Phase-conjugate wavefront generation via real-time holography in  $\text{Bi}_{12}\text{SiO}_{20}$  crystals," Optics Letters, 4 p 21, 1979.
7. B. Julesz, Foundations of Cyclopean Perception (University of Chicago Press, 1971).
8. T. Kohonen, Self-Organization and Associative Memory (Springer-Verlag, 1984).
9. G. Neuweiler, "How bats detect flying insects," Physics Today, p 34, Aug 1980).
10. D. Pepper, "Applications of Phase Conjugation," Scientific American, 254, p79, Jan 1986.
11. D. Marr, Vision (W. H. Freeman and Company, 1982).
12. M. Razmakhnin Radar Made Easy (Mir Publishers, Moscow).
13. F. Rosenblatt, Principles of Neurodynamics, (Spartan, 1962).
14. W. Stoner "Incoherent optical processing via spatially offset pupil masks," Applied Optics 17, p2454, 1978.
15. S. Wood, A. System Theoretic Approach to Image Reconstruction (University Microfilms International, Ann Arbor, MI 1978).

OPTICAL SYSTOLIC ARRAY PROCESSING USING A NOVEL  
INTEGRATED ACOUSTO-OPTIC MODULE

Chen S. Tsai, Principal Investigator

DEPARTMENT OF ELECTRICAL ENGINEERING  
UNIVERSITY OF CALIFORNIA IN IRVINE

ONR/SDI PROGRAM ON OPTICAL COMPUTING

Chen S. Tsai

Principal Investigator

Professor of Electrical Engineering

University of California, Irvine, CA 92717

TECHNICAL TITLE: OPTICAL SYSTOLIC ARRAY PROCESSING USING  
A NOVEL INTEGRATED ACOUSTOOPTIC MODULE

ABSTRACT:

During the past six-month period (July 1 to December 31, 1985) very significant progress has been made in the experimental phase of this research. Specifically, the following two experimental projects were successfully carried out:

1. Fabrication and Testing of Large Linear TIPE Microlens Array:

A linear microlens array which consists of 60 lens elements each with 30  $\mu\text{m}$  aperture and 200  $\mu\text{m}$  focal length was successfully fabricated using the titanium-indiffused proton-exchanged (TIPE) technique in a Y-cut  $\text{LiNbO}_3$  substrate. A simple matrix-vector multiplication experiment was also repeated with encouraging results using the AO modulator module that incorporated this microlens array.

2. Fabrication and Testing of Thermally-Annealed Proton-Exchanged Channel Waveguide Cutoff Modulator:

An electro-optic cutoff modulator that utilizes a single-mode thermally-annealed proton-exchanged channel waveguide in a x-cut  $\text{LiNbO}_3$  substrate has been realized for the first time. In contrast to the earlier cutoff modulators that exclusively utilized titanium-diffused channel waveguides, thermal annealing was used to provide a fine tuning on the refractive index changes that was essential in bringing the waveguides to the very edge of cutoff, and thus enabled further reduction in the drive voltage requirement. Thermal annealing was also found to greatly improve both linearity of modulation and the resistance to optical damage. For example, a device with a 2- $\mu\text{m}$  channel width and a 5 mm electrode length has demonstrated a modulation depth as high as 97% at a total voltage swing of only 7 volts and a very high linearity.

## TECHNICAL SUMMARY

### I. OBJECTIVES

The objective of this ONR/SDI-sponsored research is to advance the performance characteristics of a compact integrated acoustooptic Bragg modulator module with application to optical systolic array processing. The integrated acoustooptic module consists of a single-mode channel-planar composite waveguide, a TIPE microlens array, a SAW transducer, and a TIPE integrating lens in a Y-cut  $\text{LiNbO}_3$  substrate. The two research tasks to be performed are:

1. High-Speed Electrooptic Modulation of the Channel-Guided Light Beams, and
2. Determination of the Channel Capacity of the Integrated Acoustooptic Module.

### II. WORK PERFORMED AND RESULTS

During the past six-month period (July 1 to December 31, 1985) very significant progress has been made in the experimental phase of this research. Specifically, the following two experimental projects were successfully carried out:

#### 1. Fabrication and Testing of Large Linear TIPE Microlens Array:

A linear microlens array which consists of 60 lens elements each with 30  $\mu\text{m}$  aperture and 200  $\mu\text{m}$  focal length was successfully fabricated using the titanium-indiffused proton-exchanged (TIPE) technique in a Y-cut  $\text{LiNbO}_3$  substrate. The measured performance of this microlens array was consistent with that of the smaller arrays (with a considerably larger lens aperture and a much longer focal length) fabricated previously. A simple matrix-vector multiplication experiment involving a 2 x 2 matrix and a two-dimensional vector was also repeated with encouraging results using the resulting AO modulator module.

#### 2. Fabrication and Testing of Thermally-Annealed Proton-Exchanged Channel Waveguide Cutoff Modulator;

We have designed and fabricated the first electro-optic cutoff modulator that utilizes a single-mode proton-exchanged channel waveguide in a

x-cut  $\text{LiNbO}_3$  substrate. In contrast to the earlier cutoff modulators reported by others that exclusively utilized titanium-diffused channel waveguides, thermal annealing was used to provide a fine tuning on the refractive index changes that was essential in bringing the proton-exchanged waveguides to the very edge of cutoff, and thus enabled further reduction in the drive voltage requirement. Thermal annealing was also found to greatly improve the linearity of modulation. For example, a device with a 2- $\mu\text{m}$  channel width and a 5 mm electrode length has demonstrated a modulation depth as high as 97% at a total voltage swing of only 7 volts and a very high linearity. In addition, no optical damage has been observed after a two-hour continuous exposure of a 6328Å He-Ne laser light at an intensity as high as  $10^4$  watt/cm<sup>2</sup>. It is important to note that this simple channel waveguide cutoff modulator will take up a considerably smaller real estate and provide a higher degree of linearity in the output light intensity than the other existing modulators.

### III. CONCLUSIONS AND RECOMMENDATIONS

The experiment on the linear microlens array as described above has shown that TIPE is a viable technique for fabrication of high-performance planar waveguide microlenses, microlens arrays, and their combinations in  $\text{LiNbO}_3$  substrates using a single masking step. Using the 30  $\mu\text{m}$  lens aperture as the dimension for one basic channel, the channel capacity of the integrated AO module will be 333 per cm along the SAW propagation path. The corresponding sequential data rate for the SAW is approximately 100-Mbits/S.

In regard to the thermally-annealed proton-exchanged channel waveguide cutoff modulator explored in this research program, it should be emphasized that the modulator will take up a considerably smaller real estate and provide a higher degree of linearity in the output light intensity than the other existing modulators. It is thus viable to construct such cutoff modulators in an array configuration to facilitate multichannel operation in integrated- and fiber-optic communication, computing, and signal processing systems. For example, such a cutoff modulator array can be conveniently integrated with the AO Bragg modulator module in a  $\text{LiNbO}_3$  channel-planar composite waveguide referred to previously.

In view of the encouraging experimental results as described above, we strongly recommend continuation of the research project. The specific tasks recommended for theoretical and experimental studies are listed below:

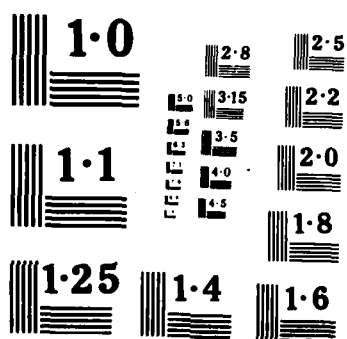
AD-A169 882 OPTICAL PROCESSING(U) DAYTON UNIV OH RESEARCH INST 6/6  
S C GUSTAFSON 31 DEC 85 UDR-TR-85-148 N00014-85-K-0479

UNCLASSIFIED

F/G 9/2

NL







#### Theoretical Study:

1. Analysis and Design of TIPE Microlenses and Lens Array
2. Analysis on AO and EO Bragg Diffractions in Channel-Planar Waveguide to Determine Ultimate Performance Such As Bandwidth, RF Drive Power, Nonlinearity, and Dynamic Range
3. Comparison Between AO and EO Modulation and/or Multiplication Schemes
4. Identification of Existing and New Architectures and Algorithms
5. Determination of Residual Noise and Error Rate

#### Experimental Study:

1. Perform AO Bragg Diffraction Experiments at Diode Laser Wavelength
2. Perform Selective Optical Computing Experiments Using the Present AO Modulation Scheme at 0.6238  $\mu\text{m}$  and Diode Laser Wavelength
3. Carry Out Optical Computing Experiments Using EO Modulation Scheme
4. Further Integration of the Basic Modulator Module With Diode Laser (or Optical Fiber) Array, Photodetector Array, and CCD Driver
5. Ultimate Realization of Integrated Optic Computer or Processor Modules and Their Performance Characterization on Accuracy, Speed, Power Consumption, Error Rate, etc.

### TECHNICAL DISCUSSION

#### 1. SINGLE-MODE TIPE MICROLENSES AND MICROLENS ARRAYS

This principal investigator's group recently developed a new and simple method which utilizes a combination of titanium-indiffusion (TI) and proton-exchange (PE) processes for formation of planar-waveguide microlenses and microlens arrays in  $\text{LiNbO}_3$  substrates.<sup>(1)</sup> Waveguide lenses have been recognized as among the basic components in Integrated Optics<sup>(2)</sup> since the inception of this now emerging technology because of the high expectations for optical communication, computing, and information processing systems to be realized in a single waveguide substrate<sup>(3)</sup>. Although a variety of planar waveguide lenses including the geodesic, chirp-grating, Fresnel, and Luneburg types had been reported in the literature, all of these lenses had thus far been fabricated in the form of single lenses only. Also, a common

characteristic of these existing lenses has been the difficulty in obtaining simultaneously a combination of all desirable lens characteristics. The TIPE microlenses and linear lens arrays fabricated thus far have demonstrated a combination of desirable properties including very short focal length, large numerical aperture (low f-number), very small focal spot size, large angular field of view, and low optical insertion loss at the 6328 Å He-Ne wavelength<sup>(1)</sup>. For fabrication of the single-mode microlenses and microlens arrays<sup>(1)</sup> the well-established TI process<sup>(4)</sup> was first applied in a Y-cut LiNbO<sub>3</sub> substrate to form a planar waveguide that supports a single TE-Mode and a single TM-mode of the lowest order. Subsequently, a masking material such as Si<sub>3</sub>N<sub>4</sub> with a designed lens contour was deposited on the TI waveguide (Figure 1a). The sample was then immersed in molten benzoic acid at 230°C for six hours. As a result of the selective proton exchange (PE)<sup>(5)</sup>, the region without the masking material had its extraordinary refractive index increased by as much as 0.11 in comparison to the remaining TI region (Figure 1b). Consequently, this TIPE region of appropriate contour will function as a planar waveguide lens. We have shown that by using the TIPE method<sup>(6)</sup> a combination of microlenses, microlens arrays, and composite lenses can be formed in the same substrate using a single masking step<sup>(1)</sup>. For example, a linear microlens array and a large-aperture integrating lens were fabricated using the TIPE method for realization of an integrated acousto-optic Bragg modulator module in a LiNbO<sub>3</sub> channel-planar composite waveguide (Fig. 2).<sup>(7)</sup>

Our ONR/SDI-sponsored research is concerned with utilization of this novel integrated AO module for optical systolic array processing. It is to be noted that "Multiplication" and "Addition" are the two basic operations for optical systolic array processing. In this case, "Multiplication" is facilitated by AO Bragg diffraction, and "Addition" by the TIPE integrating lens. Thus by pulsating the data sequences separately into the multiple input light beams and the SAW high-speed digital filtering as well as matrix-vector and matrix-matrix multiplications can be performed. A simple experiment on matrix-vector multiplication involving a 2 x 2 matrix and a two-dimensional vector has been demonstrated most recently.<sup>(7)</sup> A variety of simple and interesting experiments are being envisaged. Note that in such experiments the output data is obtained by interfacing a CCD analog shift register with the photodetector array. A number of variations to the basic configuration of Fig. 2 are also possible. For example, an array of electro-optic Bragg diffraction grating is being fabricated to replace the grating created by the SAW.

## 2. SINGLE-MODE THERMALLY-ANNEALED PROTON-EXCHANGED CHANNEL WAVEGUIDE CUTOFF MODULATOR ARRAY

There exists a popular demand for a simple optical channel waveguide modulator which possesses the following desirable attributes: 1. intensity modulation in direct proportion to the modulation voltage with a large dynamic range (in contrast to the existing modulators such as the Mach-Zehnder interferometer, directional coupler, TIR and X-modulators that exhibit a sine-squared voltage dependence of the output intensity), 2. low drive voltage and/or power requirement, 3. small real estate per modulator and thus a practical substrate size for realization of the resulting modulator array or arrays, and 4. simplicity in device configuration and fabrication process, and thus the viability of manufacturing technology involved. We have most recently succeeded in fabrication of single-mode channel waveguides in X-cut  $\text{LiNbO}_3$  substrates using the proton-exchanged (PE) process<sup>(5)</sup>. To the best of our knowledge, no report of such PE channel waveguides has appeared in the literature. We have also fabricated the cutoff modulators that utilize such PE channel waveguides and applied thermal annealing to them.. The preliminary experimental results obtained thus far have shown that this thermally-annealed cutoff modulator will provide the desirable attributes listed above.

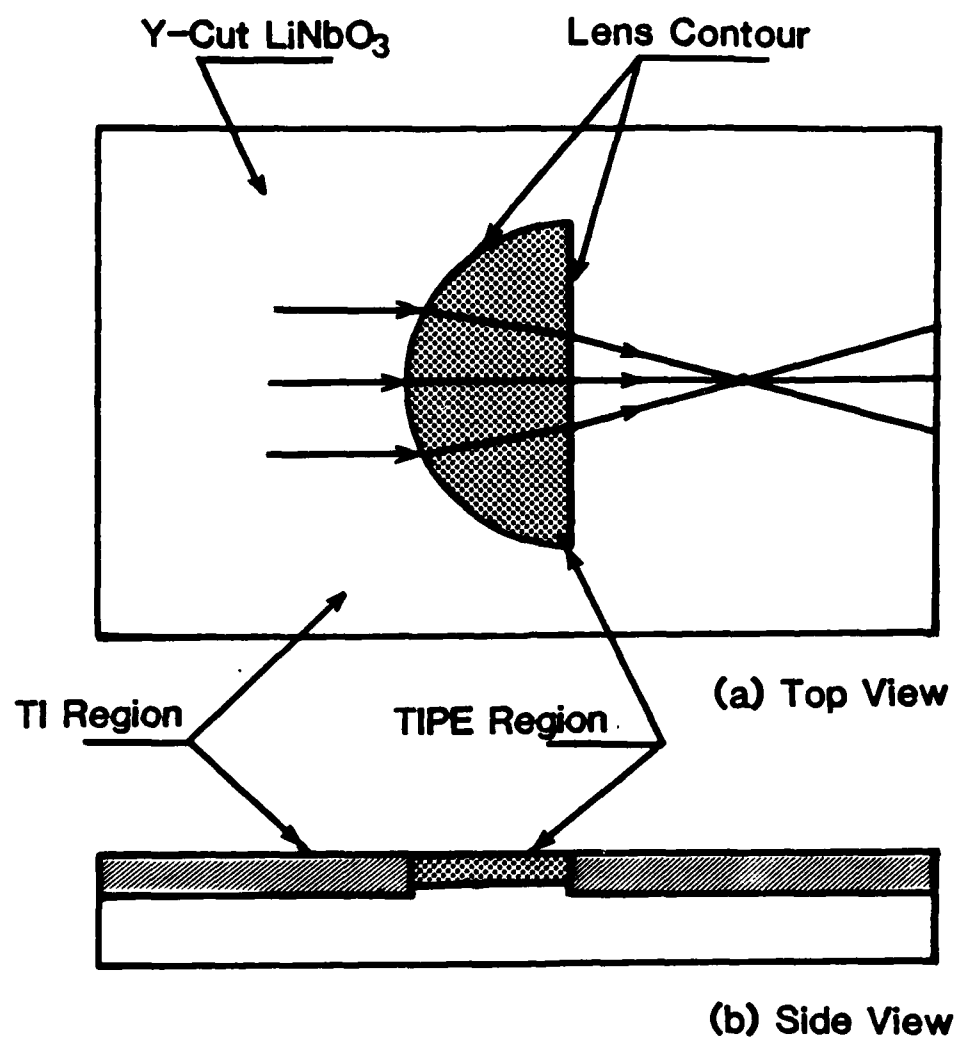
Electro-optically induced channel guiding and cutoff modulation of a light beam in GaAs,  $\text{LiNbO}_3$  and  $\text{KNbO}_3$  substrates have been previously reported. The more recent of these earlier works<sup>(8-10)</sup> all utilized titanium-indiffusion (TI) process<sup>(4)</sup> for fabrication of the waveguides in the  $\text{LiNbO}_3$  substrates. In our work the PE process<sup>(5)</sup> was employed to fabricate single-mode channel waveguides of various channel width, namely 2-, 3-, and 4- $\mu\text{m}$  in X-cut Y-propagation  $\text{LiNbO}_3$  substrates. We have designed, fabricated, and tested a number of cutoff modulators (See Fig. 3) using such PE channel waveguides and have obtained very encouraging and reproduceable results. It is to be noted that in contrast to the devices previously reported<sup>(10)</sup>, only a single uniform section of channel waveguide is involved in the present device. As a result, design and fabrication of the present device are considerably simpler. Propagation cutoff, and thus intensity modulation, are provided through electro-optical control of the extraordinary refractive index  $N_e$  using the coefficient  $r_{33}$ . For example, Plot A in Fig. 4 shows the output light intensity as a function of DC drive voltage for the device with a 2- $\mu\text{m}$

channel width and a 5mm electrode length that was measured prior to thermal annealing. It is seen that a voltage swing from -10 volts to +6 volts was required to switch the modulator from maximum transmission to cutoff, resulting in a modulation depth of 98%.

We have found it possible to bring the waveguide to the very edge of cutoff by controlling both the exchange time at a fixed exchange temperature of 245°C and the time of subsequent thermal annealing. Consequently, a further reduction in the drive voltage could be accomplished by bringing the effective refractive index of the guided mode to practically identical to the substrate index. This was facilitated by thermal annealing of the devices at 300°C for 7.0 min. in accordance with the fact that subsequent thermal annealing after the PE process would cause the index profile to shift from a stepped distribution to a graded distribution with a lower index on the surface.<sup>(11)</sup> Plot B of Fig. 5 shows the experimental demonstration of this drive voltage reduction through thermal annealing using the same cutoff modulator that was used to generate Plot A. It is seen that a 97% modulation depth was obtained at a total voltage swing of only 7 volts, namely, from -4 to +3 volts. This plot also shows a distinct improvement in the linearity of the modulation after thermal annealing. This important experimental observation may imply that the scattering loss within the modulation range became more uniform after heat treatment. Another major benefit of thermal annealing is elimination<sup>(12)</sup> of the instability in the guided mode index that was often observed in the PE waveguides.<sup>(13)</sup> This enhancement in waveguide stability through annealing was also confirmed in the PE cutoff modulator just described. Finally, in contrast to the TI devices that often suffer from the photorefractive effect and the concomittant optical damage at a relatively low light intensity, resistance to the optically induced refractive index instability in our PE devices was found to be greater. For example, no optical damage was observed even after a two-hour continuous exposure of the 6328Å wavelength He-Ne laser at a light intensity as high as  $10^4 \text{ W/cm}^2$ . This intensity threshold is at least one order of magnitude higher than that for the TI devices.

3. REFERENCES

1. D. Y. Zang and C. S. Tsai, Appl. Phys. Lett., 46, 703 (April 1985).
2. See, for example, P. K. Tien, Rev. Mod. Phys., 49, 361 (1977).
3. R. Shubert and J. H. Harris, IEE Trans. Microwave Theory Tech., KMTT-16, 1048 (1968).
4. R. V. Schmidt and I. P. Kaminow, Appl. Phys. Lett., 47, 607-608 (1982).
5. J. L. Jackel, C. E. Rice, and J. J. Veselka, Appl. Phys. Lett., 607-608 (1982).
6. M. Demicheli, J. Botineau, P. Sibillot, D. B. Ostrowsky, and M. Papuchon, Opt. Commun., 42, 101-103 (1982).
7. C. S. Tsai, D. Y. Zang, and P. Le, Appl. Phys. Lett., Vol. 47, pp. 549-551 (Sept. 1985).
8. Osamu Mikan and Akira Ishida, Japan J. Appl. Phys., Vol. 12, pp. 1294-1295 (1973).
9. A. Neyer and W. Sohler, Appl. Phys. Lett., Vol., 45, pp. 840-842 (1979).
10. P. R. Ashley and W. S. C. Chang, Appl. Phys. Lett., Vol., 45, pp. 840-842 (1984).
11. M. DeMicheli, J. Botineau, S. Neveau, P. Sibillot, D. B. Ostrowsky, and M. Papuchon, Optics Letters, Vol. 8, No. 2, pp. 113-115 (1980).
12. J. L. Jackel and C. E. Rice, SPIE, Vol. 460, pp. 43-48 (1984).
13. Alfiedo Yi-Yan, Appl. Phys. Lett., Vol. 42, pp. 633-635 (1983).



**Fig. 1 Planar Waveguide Lens In LiNbO<sub>3</sub> Formed By Titanium Indiffused Proton Exchanged (TIPE) Technique**

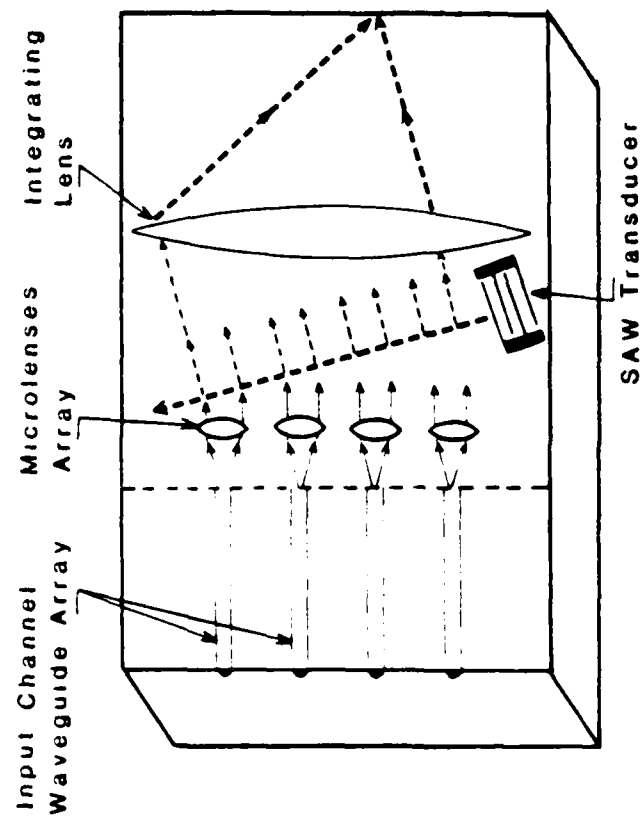


Fig. 2 Acoustooptic Bragg Diffraction In A  $\text{LiNbO}_3$   
Channel-Planar Composite Waveguide

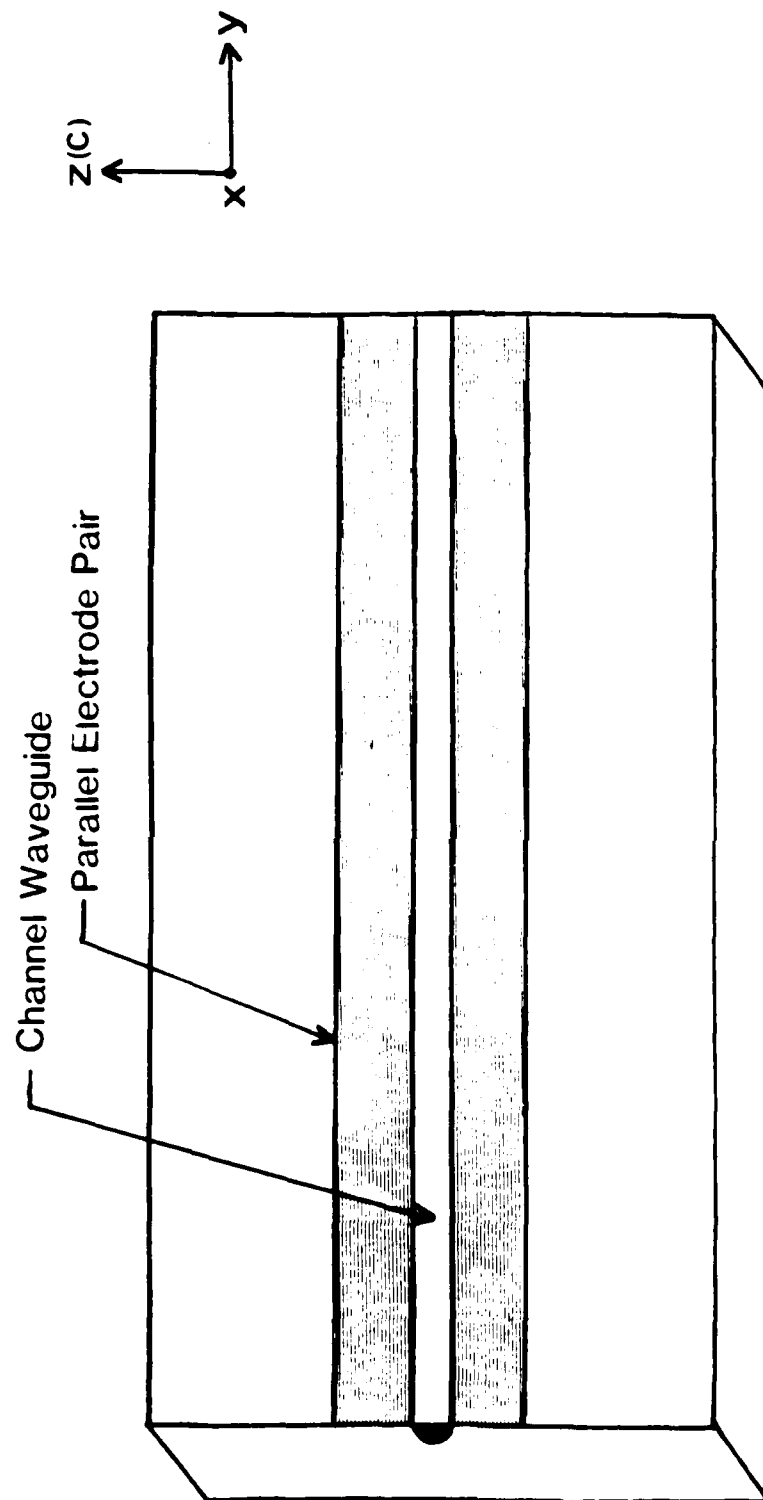


Fig. 3 Single-Mode Proton-Exchanged Channel Waveguide  
Cut-Off Modulator In X-Cut Y-Propagation  $\text{LiNbO}_3$



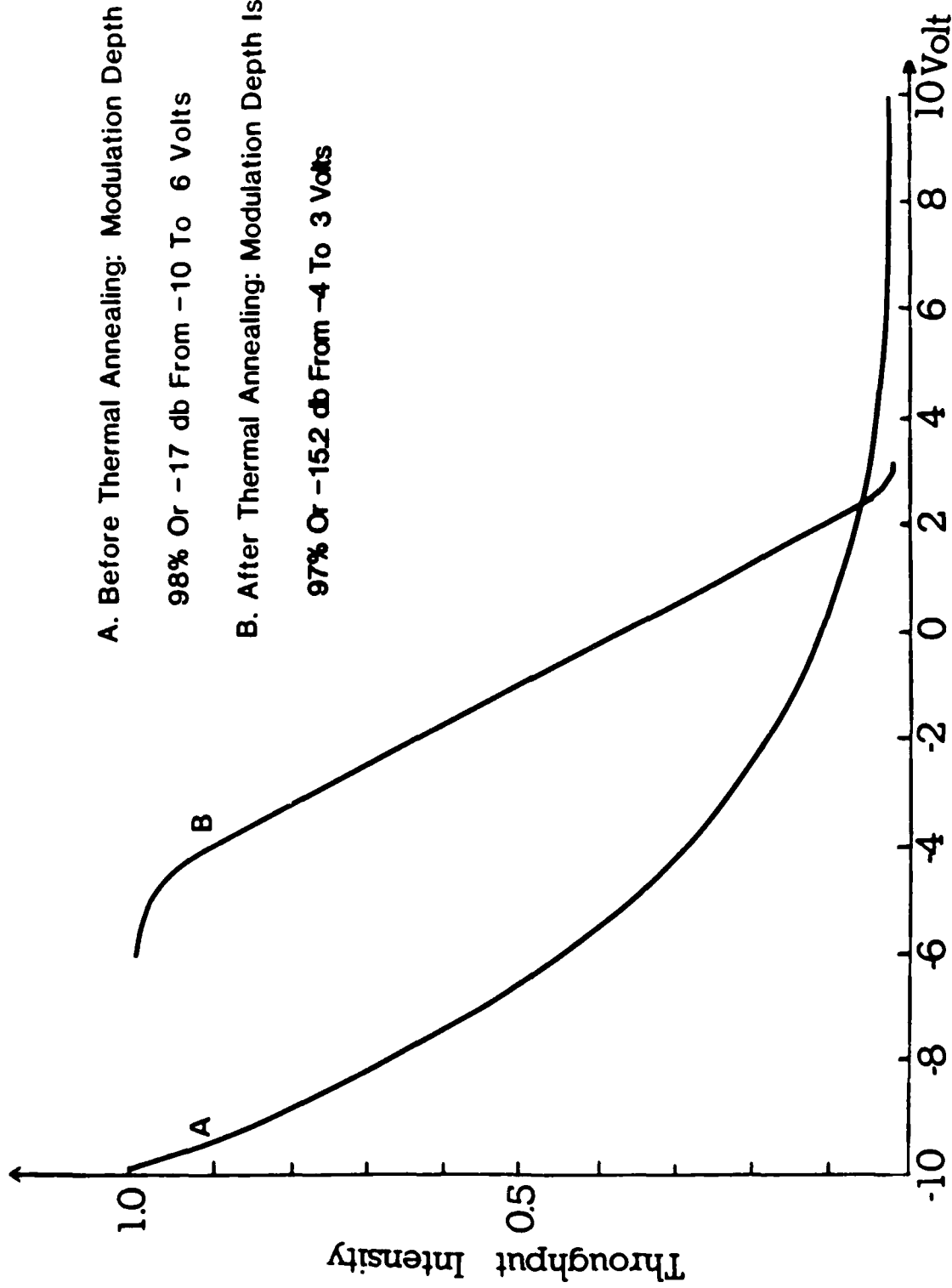


Fig. 4 Modulation Curve Of The Device With A 2  $\mu$ m Channel Width And A 5 mm Interaction Length

OPTICAL COMPUTING COMPONENTS: FUNDAMENTAL ISSUES

Armand R. Tanguay, Jr., Principal Investigator

UNIVERSITY OF SOUTHERN CALIFORNIA

**OPTICAL COMPUTING COMPONENTS:  
FUNDAMENTAL ISSUES**

**FINAL TECHNICAL REPORT**

Submitted To:

University of Dayton Research Institute  
300 College Park Avenue  
Dayton, Ohio 45469  
Attn: Dr. Stephen Gustafson

Submitted By:

Dr. Armand R. Tanguay, Jr.  
Optical Materials and Devices Laboratory  
Departments of Electrical Engineering  
and Materials Science, and  
Center for Photonic Technology  
University of Southern California  
Los Angeles, California 90089-0483

**ABSTRACT**

This research program addressed fundamental performance issues and tradeoffs of a variety of optical processing and computing components, such as spatial light modulators, photorefractive volume holographic optical elements, and threshold arrays. Such issues were examined in the context of the incorporation of such components in optical processing and computing systems, in order to establish predicted performance boundaries based where possible on physical principles.

# **OPTICAL COMPUTING COMPONENTS: FUNDAMENTAL ISSUES**

## **TECHNICAL SUMMARY**

### **I. OBJECTIVES**

This program addresses a number of critical issues that potentially define and delimit the implementation of optical processing and computing systems.

Recently, tremendous advances have occurred in the development of potential algorithms and architectures for dimensionally large computational problems that are particularly well suited to parallel processing. These suggested algorithms and architectures have for the most part, however, been developed without careful attention to constraints imposed by fundamental physical effects that bound implementable device performance.

The past few years have also witnessed dramatic growth in available component technologies. These include advances in bulk wave acoustooptic modulators, surface acoustic wave devices, novel one- and two-dimensional spatial light modulators, new and improved photorefractive materials for volume holographic optical elements, multiple quantum well and superlattice structures, and bistable optical devices.

Recently, several research groups have begun to examine the implications of the at times conflicting requirements placed on optical information processing and computing components by proposed computational architectures and their associated algorithms, and also on the architectures by available components [1, 2, 3, 4]. This type of analysis proceeds by examining the ultimate limits of system performance achievable by various architectural arrangements of presently available as well as potentially available components. These limits derive from power dissipation, thermal noise, input power, quantum uncertainty, material nonlinearity, and detection signal-to-noise considerations. The perspective taken here is that accelerated research and development of optical information processing and computing systems will be most effective if continued research is focused on physically realizable components with tractable technological hurdles.

During the research program, we proposed to study a wide range of potential algorithms, architectures, and components with the goal of identifying eventual performance boundaries that are based on fundamental physical limitations. The program focus was such as to elucidate significant avenues of opportunity for both optical

processors/computers and hybrid optical interconnect/VLSI dynamically reconfigurable machines.

## II. DESCRIPTION OF THE RESEARCH EFFORT

The research performed under this contract consisted of two principal components, as outlined in the statement of work. The first involved technical interactions with other subcontractors at several individual and group meetings (including the 1985 Annual Meeting of the Optical Society of America (Washington, D. C., October, 1985), with participation in the UDRI Program Review). These interactions were focused on elucidating optimized goals for continued and accelerated research in optical processing and computing algorithms, architectures, components, and materials. The second principal component comprised research on fundamental physical limitations to device performance, with results described in this section.

The scope of the effort was limited at the outset to the study of problem definition and feasibility of approach. Nonetheless, several important results were obtained even in this preliminary phase, as outlined below.

Four principal areas of investigation were identified as of critical importance to the eventual implementation of optical processing and computing systems: spatial light modulators (both optically and electrically addressed), dynamically programmable volume holographic optical elements, threshold arrays, and absolute limits imposed by the physics of computation.

The investigation of spatial light modulator limitations focused on fundamental effects that are technology-independent. Of primary importance is consideration of the quantum fluctuations characteristic of incident illumination, and the concomitant effects of such fluctuations on the tradeoffs allowable among spatial resolution, dynamic range, and frame rate. It was found that for the case of binary operations, such limitations are quite similar to those encountered in traditional electronic computing machines, with the exception that the energy cost per photon is typically larger than the energy cost for transport of the corresponding electron. On the other hand, the case of analog (high dynamic range) computation presents an interesting situation in which quantum fluctuations play a fundamentally important role in establishing minimum performance boundaries [4]. These performance boundaries are surprisingly positioned, and thus have implications for device design that have not heretofore been taken into consideration.

We have now extended this concept to include the cases of mutual coherence/incoherence between source and detector, and are in process of applying the principles of coherent detection from communication theory to the optical processing/computing case of area-detection. It should be pointed out that the process

of incoherent-to-coherent conversion is intrinsically a detection, with the spatial light modulator itself taking the place of the detector. A second equivalent situation occurs when the input signal is detected either electronically or optically on a detector (either serial or parallel), and is subsequently sequenced onto the SLM by either electronic or optical encoding techniques.

The use of photorefractive materials for the fabrication of dynamically programmable volume holographic optical elements has received considerable attention for a wide range of applications in optical processing and computing. It is therefore of considerable interest to examine potential fundamental limitations of this technology that are not material-dependent. This investigation is quite timely, as intensive studies of photorefractive materials have been undertaken recently, and such studies are time-consuming, difficult to perform reliably, expensive, and in most cases somewhat investigator-dependent. During the contract period, we have developed an extensive interactive computer model of the holographic grating formation process, with capability for inclusion of a wide range of material characteristics, exposure parameters, physical mechanisms, polarization effects, and volume diffraction characteristics. This model has allowed us to explain all observed characteristics of photorefractive materials such as bismuth silicon oxide, and to further predict a number of fundamental constraints. Several theorems have been proven regarding the nature of diffraction from birefringent phase gratings in the Bragg regime, particularly with regard to the resultant polarization states (which are of considerable interest from signal-to-noise considerations for orthogonalization of the incident and diffracted beams).

Recently, we have shown that the inclusion of constant velocity gratings, which have received considerable attention for application to phase conjugate image amplification, is likely inapplicable to the cases desired for most optical processing and computing applications. In fact, we have shown that as the modulation index of the writing beams (i. e. of the stored grating) approaches unity, the stationary grating case provides significantly larger diffraction efficiency than the constant velocity grating case. This result has significant implications for the utilization of transient gratings in system designs incorporating photorefractive materials.

Threshold arrays are needed in optical processing and computing for a wide range of applications, with a concomitant wide range of required performance parameters. In particular, while bit-oriented optical computers may require ultra-low power switching arrays operating at relatively high frame rates, associative memory processors may have considerably relaxed requirements on switching energy per pixel and frame rate. During the contract period, we have examined the power consumption requirements of all reported threshold array elements, and find that they are in all cases to date too high for

potential inclusion in a bit-mapped parallel digital optical computer that is competitive with currently available electronic technology. A key result is that such performance boundaries focus increased importance on the complexity of interconnections designed into each stage of the processor. The higher the level of interconnection complexity per iteration (bit plane), the greater the chance that such systems will prove feasible.

During the contract period, we have examined in considerable detail the implications for optical processing and computing of recent assertions concerning negligible computational energy consumption in reversible computational elements [5,6,7]. Although we are in full agreement that such systems are in fact theoretically achievable, an important conclusion has emerged from our study. A basic tenet of the reversible machine is an adiabatic approach to each binary decision. This implies not only a large uncertainty as to the time constant over which the decision is registered, but also intrinsically large time constants for even a minimum decision time. Therefore, it can be stated that the energy cost of computation so often quoted previously is not in fact inherent in the nature of computation, but is in fact inherent in deterministic computations (in which the decision to be rendered is stabilized against uncertainties in the shortest possible time). Thus the energy cost of computation can only be calculated within the context of a particular architecture, and within carefully stated performance goals for overall processor speed.

### III. CONCLUSIONS AND RECOMMENDATIONS

From the results of this preliminary research effort, we conclude that numerous performance limitations will accrue to eventual optical processing and computing systems that derive principally from inherent limitations in materials, devices, and system architectures. In particular, a number of such limitations were derived for spatial light modulators, photorefractive materials and

dynamically programmable volume holographic optical elements, and threshold arrays. In addition, the energy of computation assignable to a given architecture implemented with a given set of components can be directly determined from the stated performance goals of the overall system.

The principal recommendation of this study effort is straightforward. As a community of optical processing/computing scientists and engineers, we must pay strict attention to fundamental limitations in the generation of optimistic system performance expectations. In addition, much work remains to be done in the elucidation of these fundamental performance boundaries. Continued interaction between device/materials scientists and algorithm/architecture scientists will be crucial to establishing implementable systems concepts with a viable component technology base.

#### IV. REFERENCES

1. P. W. Smith, "On the Physical Limits of Digital Optical Switching and Logic Elements", Bell Syst. Tech. Jour., 61, 8, 1975-1993, 1982.
2. J. W. Goodman, F. J. Leonberger, S-Y. Kung, and R. A. Athale, "Optical Interconnections for VLSI Systems", Proc. IEEE, July, 1984, 850-866.
3. D. Psaltis, "Two-Dimensional Optical Processing Using One-Dimensional Input Devices", Proc. IEEE, July, 1984, 962-974.
4. A. R. Tanguay, Jr., "Materials Requirements for Optical Processing and Computing Devices", Optical Engineering, 24(1), 002-018, 1985.
5. R. P. Feynman, "Quantum Mechanical Computers", Optics News, 11(2), 11-20, 1985.
6. C. H. Bennett, "Logical Reversibility of Computation", IBM Jour. Res. Dev., 6, 525-532, 1979.
7. E. Fredkin and T. Toffoli, "Conservative Logic", Int. J. Theor. Phys., 21, 905-940, 1982.



END

DTIC

8-86